NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTOMATING INFORMATION ASSURANCE FOR CYBER SITUATIONAL AWARENESS WITHIN A SMART CLOUD SYSTEM OF SYSTEMS

by

Kuan Wei Edmund Teo

March 2014

Thesis Advisor:	Deborah E. Goshorn
Co-Advisor:	Gary W. Parker

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
AUTOMATING INFORMATION ASSURANCE FOR CYBER SITUATIONAL AWARENESS WITHIN A SMART CLOUD SYSTEM OF SYSTEMS

5. FUNDING NUMBERS

6. AUTHOR(S) Kuan Wei Edmund Teo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

In a world in which data is being generated in increasing large volumes and is easily accessible to multiple users in a cloud environment, there is a need to maintain situational awareness and information assurance of the data, ensuring the data is being monitored for vulnerabilities. This is especially crucial for military operations where the information being used to support the mission is confidential and readily available throughout the mission. It is essential to maintain the integrity of that information. The need is even more critical when data is being used to help save lives in natural disaster situations.

A trio system concept within an enterprise/cloud network is developed in this research to provide situational awareness and command and control abilities to users, for detecting possible cyber attacks on network and computing resources, and maintaining confidentiality, integrity, and availability of critical data within the network.

A systems engineering approach was used to develop and propose the solution to ensure information assurance and cyber situational awareness within a smart cloud of system of systems. This thesis provides system diagrams of the proposed architecture focusing on one of the systems using IDEF0 diagrams, and a feature matrix to demonstrate the concept of Detect, Identify, Predict, and React model. A proof-of-concept experiment for the Identify model is discussed.

14. SUBJECT TERMS Automating Information Assurance, Cyber Situational Awareness, DIPR Model, Smart Cloud System of Systems, Cyber Sensors, Systems Engineering, Data Science

15. NUMBER OF PAGES
161

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

AUTOMATING INFORMATION ASSURANCE FOR CYBER SITUATIONAL AWARENESS WITHIN A SMART CLOUD SYSTEM OF SYSTEMS

Kuan Wei Edmund Teo
Senior Engineer, Defence Science and Technology Agency
B.Eng., Nanyang Technological University, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2014

Author: Kuan Wei Edmund Teo

Approved by: Deborah E. Goshorn
Thesis Advisor

Gary W. Parker
Co-Advisor

Clifford Whitcomb
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In a world in which data is being generated in increasing large volumes and is easily accessible to multiple users in a cloud environment, there is a need to maintain situational awareness and information assurance of the data, ensuring the data is being monitored for vulnerabilities. This is especially crucial for military operations where the information being used to support the mission is confidential and readily available throughout the mission. It is essential to maintain the integrity of that information. The need is even more critical when data is being used to help save lives in natural disaster situations.

A trio system concept within an enterprise/cloud network is developed in this research to provide situational awareness and command and control abilities to users, for detecting possible cyber attacks on network and computing resources, and maintaining confidentiality, integrity, and availability of critical data within the network.

A systems engineering approach was used to develop and propose the solution to ensure information assurance and cyber situational awareness within a smart cloud of system of systems. This thesis provides system diagrams of the proposed architecture focusing on one of the systems using IDEF0 diagrams, and a feature matrix to demonstrate the concept of Detect, Identify, Predict, and React model. A proof-of-concept experiment for the Identify model is discussed.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
C2	Command and Control
CADSA	Cyber Attack Detection Situational Assurance
CCE	Common Configuration Enumeration
CPE	Common Platform Enumeration
CSAC2	Cyber Situational Awareness and Command and Control
CTO	Chief Technology Officer
DCO	DoD Cyber Operations
DGO	DoDIN Global Operations (formerly DoD GIG Operations)
DIPR	Detect, Identify, Predict and React
DLP	Data Loss Prevention
DoD	Department of Defense
DoDIN	DoD Information Network
EMMI	Energy, Material, Material Wealth, and Information
GIG	Global Information Grid
HADR	Humanitarian Assistance and Disaster Relief
HTML	HyperText Markup Language
IA	Information Assurance
IAI	Israel Aerospace Industries
IASA	Information Assurance Situational Awareness
IDEF0	Integrated Computer Aided Manufacturing (Icam) DEFinition for Function Modeling
IP	Internet Protocol
IPR	Identify, Predict and React
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
ISR	Intelligence Surveillance and Reconnaissance
IW	Information Warfare
JIFX	Joint Field Experimentation
JPEG	Joint Photographic Experts Group

JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LAN	local area network
MAC	media access control
MS	Microsoft
NAC	network access control
NORAD	North American Aerospace Defense Command
NPS	Naval Postgraduate School
OPNAV	Office of Chief of Naval Operations
ONR	Office of Naval Research
OUSD (AT&L)	Office of the Under Secretary of Defense, Acquisition, Technology, & Logistics
OUSD (P)	Office of the Under Secretary of Defense, Policy
RAM	random access memory
REST	Representational State Transfer
RSS	rich site summary
SA	situational awareness
SAF	Singapore Armed Forces
SoS	system of systems
SSL	secure sockets layer
S&T	Science and Technology
SYN	synchronize
TCP	transmission control protocol
TLS	transport layer security
USN	United States Navy
USNORTHCOM	United States Northern Command
VADM	Vice Admiral
VPN	virtual private network
XCCDF	eXtensible Configuration Checklist Description Format
XML	Extensible Markup Language

EXECUTIVE SUMMARY

A new realm of threat has emerged with the dawn of cyber warfare; data is being generated in increasingly huge amounts, making the management and monitoring of data difficult. Cyber attack methodology has also become more sophisticated. Ensuring data has not been compromised and maintaining its confidentiality, integrity, and availability is crucial for military operations, especially when data is being used to save lives in humanitarian assistance and natural disaster relief operations. This thesis proposes a system of systems engineering methodology for integrating and configuring an intelligence automation system for the purpose of enabling cyber command and control and cyber situational awareness.

A trio of intelligence automation systems, namely Cyber Situation Awareness and Command and Control (CSAC2), Cyber Attack Detection Situation Awareness (CADSA), and Information Assurance Situation Awareness (IASA), was proposed to achieve the purpose of providing automated information assurance for cyber situation awareness of data, including the assurance of data confidentiality, integrity, and availability. A proof of concept using the Identify software of the Detect, Identify, Predict, and React (DIPR) model of the IASA system was used to automate the information assurance of a target file, which is in the data at rest state, against four test cases. The four test cases are normal operation, loss of confidentiality, loss of integrity and loss of availability of a monitored target of interest. The proof of concept successfully analyzed and updated the feature matrix produced by the Detect software of DIPR model for each of the four test scenarios. Further research can be done to improve the usage of other file characteristics, such as monitoring candidates, studying the effectiveness of monitoring data in other data states, and expanding experimental testing by integrating CSAC2 and CADSA into the test.

A systems engineering approach was used to explore the possible solution to the problem statement focusing mainly on the IASA system. A systematic analysis of the requirements to automate information assurance in the areas of data confidentiality, integrity, and availability for cyber situational awareness within a smart cloud system of

systems was performed. After going through the systems engineering process and generating an architecture with an analysis of alternatives, this thesis implements and describes an instantiated proof-of-concept system of a portion of the IASA system, with documented test and evaluation results.

Making use of the Detect, Identify, Predict, and React (DIPR) model, the IASA system is able to perform the task of ensuring information assurance by monitoring, analyzing, and comparing the file characteristics, such as the timestamp and file size of the monitored target. This thesis focused on data in the state of at rest/stored, but the concept can be expanded to other data states such as in transit or in use by applying different types of cyber sensors. The concept can be further expanded with the implementation of CSAC2 and CADSA systems in the network, thereby providing synergy between the applications and producing a robust setup. This trio system of systems will automate generation of alerts to operators and activate cyber controls to close ports or encrypt files in the event of cyber attacks or degradation in information assurance.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my advisors, Dr. Deborah E. Goshorn, for her unwavering guidance, and Mr. Gary W. Parker, for his relentless support for this thesis. Without their valuable feedback and persistent help this thesis would not have been possible. Both experts have offered me a valuable lesson in the field of cyber security and information assurance in cloud environment.

I would also like to thank those who have helped me in one way or another during my preparation of this thesis.

Last, but not least, I am thankful to and fortunate enough to get constant support, love, and encouragement from my wife, Karen Chen, who helped to manage the house and kids so well during my course of study in NPS; she is also my source for delicious food. I am thankful also to my beloved beautiful kids, Jazzelle and Jareth, for bringing their laughter and joy into my life.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

As an introduction to this thesis, this chapter first explains the reason for selecting this thesis topic, which is derived from both a project on Smart Cloud System of Systems and direct discussions of the cybersecurity issues on Smart Cloud System of Systems with Vice Admiral (VADM) Jan Tighe, currently Commander of 10th Fleet Cyber Command. This chapter then reviews additional stakeholder input for selecting the operational scenario for this thesis application. This chapter then presents the resulting proposed concept of the thesis and the capability gap that the concept fills. Finally, this chapter concludes with a high-level overview of the thesis organization.

A. SMART CLOUD SYSTEM OF SYSTEMS

As part of the Systems Engineering curriculum at Naval Postgraduate School (NPS), a project-based, three-course sequence is required that covers Engineering Systems Conceptualization, Engineering Systems Design, and Engineering Systems Implementation & Operation (Goshorn 2013). There were several projects to choose from, and the author chose to work with the Smart Cloud System of Systems (SoS) with Dr. Deborah Goshorn. This is based on over fifteen years of research on intelligence automation and network-centric system of systems (Goshorn 2010). Figure 1 depicts the Smart Cloud System of Systems model used for representing a network-centric system of systems. The Smart Cloud SoS is made up of a major cloud node and several “mini cloud” nodes. The “mini clouds” represent nodes at tactical sites or at sites in which an operator and/or analyst is using data from the cloud to make a decision. The main cloud node represents a set of enterprise cloud nodes that perform further analysis of the data and is in charge of routing the right data to the right operator/analyst.

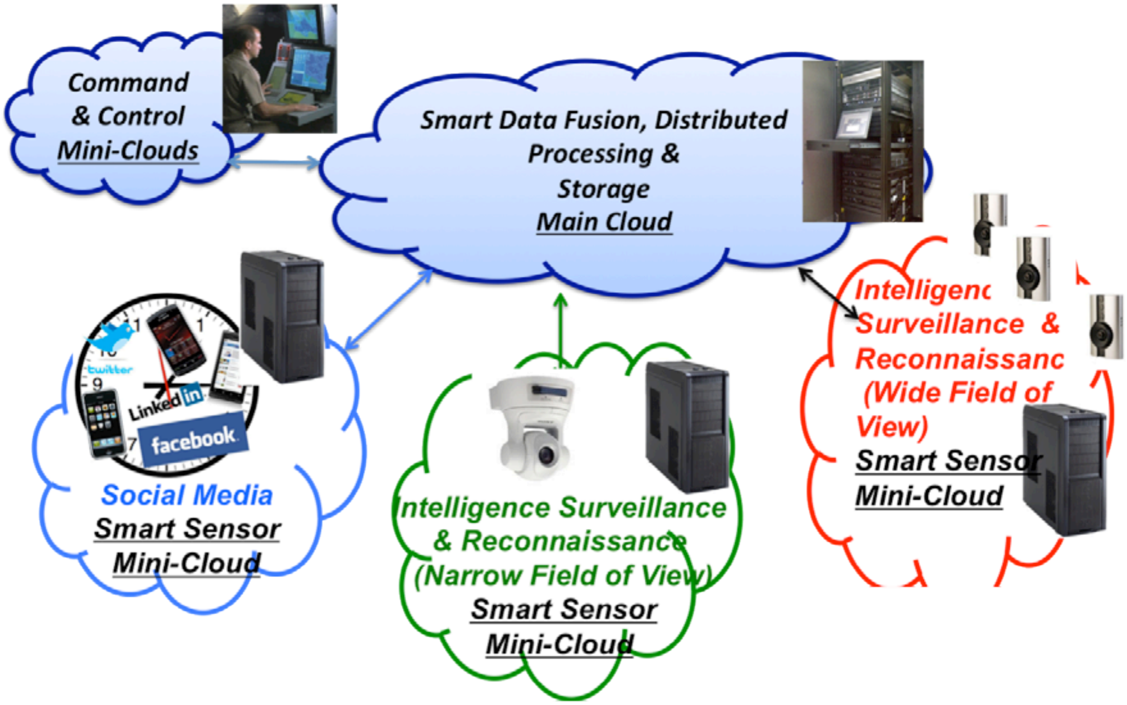

Command & Control Mini-Clouds
Smart Data Fusion, Distributed Processing & Storage Main Cloud
Intelligenc Surveillance & Reconnaissanc (Wide Field of View) Smart Sensor Mini-Cloud
Social Media Smart Sensor Mini-Cloud
Intelligence Surveillance & Reconnaissance (Narrow Field of View) Smart Sensor Mini-Cloud
twitter
Linked in
facebook

Figure 1. Smart cloud system of systems (after Goshorn 2013).

In terms of mini-clouds, there are two kinds: smart sensor mini clouds, and command and control mini clouds. The smart sensor mini clouds collect and analyze data from sensors in multiple domains, such as social media and the physical domain, creating sensor feed data from the Intelligence Surveillance and Reconnaissance (ISR) systems. In addition to the raw data feed (e.g., video feed or social media feed), these mini-clouds also perform a first-pass level of automation and provide output in a standardized alert data file format to the main cloud. The main cloud further fuses the data alerts using automation software, stores the data on the main cloud, and finally distributes the right data to the right user.

Finally, the command and control "mini clouds" serve as the hubs to gather input from the operators and analysts, and also to provide situational awareness (SA) in the form of alerts. The operators input and configure the entire smart cloud system of systems based on what types of situational awareness alerts they want to receive (Goshorn 2013).

B. DISCUSSION WITH VADM JAN TIGHE ON CYBER SA APPLIED TO SMART CLOUD SYSTEM OF SYSTEMS

Figure 2. VADM Tighe discusses the need for cyber situational awareness within Smart Cloud System of Systems at the Distributed-GIG Intelligence Automation Systems (DGIAS) Lab for Military and Homeland Security (author in far right of picture) (from Ammon 2013).

As a member of the Smart Cloud System of Systems project, it was an honor to receive direct stakeholder input from our NPS Interim President, VADM Jan Tighe, who was also acting as director, Decision Superiority OPNAV N2N6F4 at the time, and is now currently serving as Commander, Fleet Cyber Command/Commander, 10th Fleet. During this guest lecture, VADM Tighe discussed information dominance and information warfare (IW) and shared her experiences as a senior IW officer.

In this discussion, the need for capturing information assurance of data within the smart cloud system of systems was relayed by VADM Tighe. As further discussed with the primary advisor of this thesis, this means it is important to monitor the information assurance of the data in its different states of “data at rest,” “in transit,” and “in use” as it moves throughout the smart cloud system of systems. More background on information assurance is provided in Chapter II.

After hearing the importance of being able to monitor information assurance within a cloud environment, and realizing all of the important systems connected to the cloud that could end up with fatal ramifications if the information is not assured, the author developed this concept for solving information assurance behavior analysis based on this need and the existing intelligence automation capabilities from prior research (Goshorn 2010).

As an outcome of this lecture and discussion, this thesis and one other were conceived, as documented in an article published by NPS (Ammon 2013).

C. ADDITIONAL STAKEHOLDER INPUT ON CYBER SA OF SMART CLOUD SYSTEM OF SYSTEMS

This section discusses the stakeholder input received by the author both directly and indirectly, which directed this thesis topic. By participating in the Smart Cloud System of Systems project, the author had access to discussions with several stakeholders (Goshorn, 2013). Direct input from U.S. Navy (USN) stakeholders included Vice Admiral (VADM) Jan Tighe, U.S. Navy, along with the Chief Technology Officer (CTO) of United States Northern Command/North American Aerospace Defense Command (USNORTHCOM/NORAD) Science & Technology (S&T) office. Indirect input was derived from the Office of Naval Research (ONR) along with stakeholders within Singapore, such as the Singapore Armed Forces.

1. USN Stakeholder Input

As discussed in the previous section, the need for providing information assurance of data within a cloud environment was discussed by VADM Tighe. This discussion is what sparked the development of the capability architected in this thesis.

2. NORTHCOM/NORAD Stakeholder Input

As part of being in the Smart Cloud project within the SE3201 Engineering Systems Conceptualization course, stakeholder Dr. Hal Moore, CTO of NORAD-

NORTHCOM/ S&T Office, discussed operational scenarios that are of importance to him. In this discussion, he relayed the need to be prepared for cyber attacks on the smart cloud system of systems (SoS) when the SoS is deployed in a tactical environment for the purpose of supporting a Human Assistance/Disaster Relief (HA/DR) operation. More specifically, understanding the validity and integrity of the data alerts that are sent to the decision maker at the Command and Control "mini cloud" was of upmost interest.

This discussion not only confirmed the need to have information assurance of data to ensure its confidentiality, integrity, and availability, it also scoped which operational scenario the thesis should use: the HA/DR scenario.

3. Office of Under Secretary of Defense, Policy Stakeholder Input

As part of the smart cloud system of systems project, access to additional stakeholders at the Joint Field Experimentation (JIFX) was made available.

Figure 3. Mr. Al Miller (third from left) from USD(P) discusses the need for interoperable alerts within a smart cloud system of systems (author on far left.)

For example, as captured in Figure 3, the Science and Engineering Advisor, Mr. Al Moore, for Policy Integration from the Office of Secretary of Defense, Policy (OUSD(P)), discussed the importance of generating alerts that are interoperable within the smart cloud system of systems.

This requirement was taken into consideration as this thesis concept system does output its alerts using an interoperable standard, as discussed in Section B of Chapter II.

4. Office of Under Secretary of Defense, Acquisition, Technology, and Logistics Stakeholder Input

As part of the smart cloud system of systems project, access to additional stakeholders at the JIFX was made available.

Figure 4. Mr. Richard Marchant (far right) from OUSD (AT&L) discusses the need to perform test and evaluation of new capabilities (author second from right).

For example, as captured in Figure 4, Mr. Richard Marchant from the Office of Under Secretary of Defense, Acquisition, Technology, and Logistics (OUSD (AT&L)) discusses the need for performing test and evaluation of new capabilities that trace to requirements.

This thesis includes a traceability matrix in Chapter III that maps the proposed system’s capability to originating requirements. In addition, this thesis provides a preliminary test and evaluation of the capability’s proof-of-concept.

5. Office of Naval Research Stakeholder Input

In SE3201 Engineering Systems Conceptualization course, the primary advisor, Dr. Goshorn presented the need for an intelligence automation infrastructure within the smart cloud system of systems. This infrastructure uses the Detect-Identify-Predict-React (DIPR) model that is described in previous work and is reviewed in Chapter II (Goshorn 2010).

Additionally, the cyber application of this DIPR intelligence automation framework is considered under current research for Office of Naval Research for Code 30, ISR Thrust Program Manager. In this case, stakeholder input is received indirectly from the advisor’s research with the Office of Naval Research for a framework where learning can be achieved based on cyber sensor deployment on the smart cloud system of systems. Figure 5 explains the big picture setup of using DIPR to automate cyber situational awareness of smart cloud system of systems, called “blue team” cyber behavior intelligence automation. “Red team” cyber behavior intelligence automation refers to automating when and how someone should perform a cyber attack on the smart cloud system of systems. Further background on how to apply the DIPR Intelligence automation framework to do this from a blue team perspective performing active cyber defense is found in previous research (Jurjonas 2012). Background on applying cyber intelligence automation from a red team perspective is provided in other research in addition to this thesis (Deptula 2013).

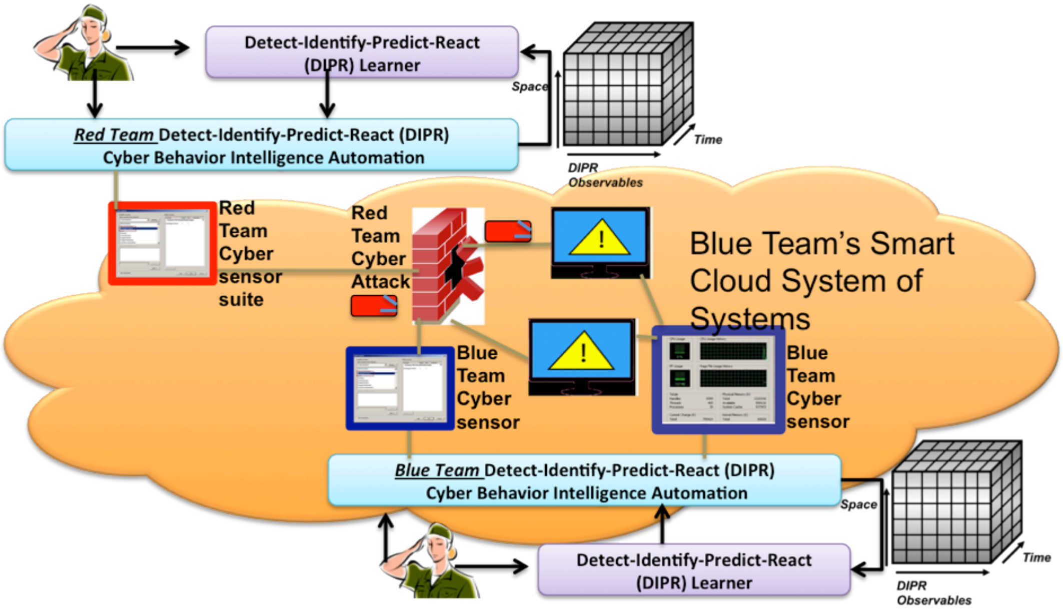

Detect-Identify-Predict-React (DIPR) Learner
Red Team Detect-Identify-Predict-React (DIPR) Cyber Behavior Intelligence Automation
Space
Time
DIPR Observables
Red Team Cyber sensor suite
Red Team Cyber Attack
Blue Team's Smart Cloud System of Systems
Blue Team Cyber sensor
Blue Team Cyber sensor
Blue Team Detect-Identify-Predict-React (DIPR) Cyber Behavior Intelligence Automation
Space
Time
Detect-Identify-Predict-React (DIPR) Learner
DIPR Observables

Figure 5. Research with the Office of Naval Research applies the Detect-Identify-Predict-React (DIPR) intelligence automation framework to cyber situational awareness within a smart cloud system of systems (Goshorn 2013).

6. Singapore Armed Forces and Other Singaporean Stakeholder Input

Finally, as part of this stakeholder analysis review, a preliminary research review on potential stakeholder within Singapore was conducted. From this research, the most significant stakeholder determined was the Singapore Armed Forces and their new Cyber Defence Operations Hub. On June 29, 2013, Defence Minister Dr. Ng Eng Hen (Figure 6) announced the setting up of the hub in order to “fight increasingly prevalent cyber attacks” (Lim 2013). Thus, the need for cyber SA is prevalent within Singapore’s military.

Figure 6. Defence Minister Dr. Ng Eng Hen announces the new Cyber Defence Operations Hub on June 29, 2013 (Lim 2013).

Additional Singaporean stakeholders include research and development organizations such as Israel Aerospace Industries (IAI), who launched a cyber research and development center in Singapore, in cooperation with Singapore's Economic Development Board (The Jerusalem Post 2014). A photograph supporting the launch of the Singapore IAI R&D Center is captured in Figure 7.

Figure 7. Israel Aerospace Industries (IAI) cyber research and development center in Singapore (Jewish Business News 2014).

D. HUMANITARIAN ASSISTANCE/DISASTER RELIEF OPERATION

A humanitarian assistance/disaster relief (HA/DR) operation was the scenario selected by various stakeholders for the smart cloud system of systems project. HA/DR operations support the nation or region under a disaster in order to minimize deaths and suffering in the affected public. Also, HA/DR operations are supposed to help improve security and stability in the region after a disaster. Finally, HA/DR operations help transition the region to prepare for long-term recovery. Example activities that occur during an HA/DR operation include providing technical aid to the region, building disaster relief warehouses, maintaining emergency operation centers, building shelters and potentially building fire stations. Finally, during these operations, essentials such as food, water, and medication are delivered and distributed throughout the region hit by the disaster (USSOUTHCOM 2014).

Figure 8 depicts how the smart cloud system of systems can be deployed in a HADR scenario for the purpose of detecting terrorist attacks in an area where there are multiple crowds forming. This project assumed there is an operation going on where there are both social media sensors and camera sensors (on both unmanned aerial vehicle (UAV) platforms and tower-based ground platforms) that are performing surveillance of an area where aid is being provided to the masses of people.

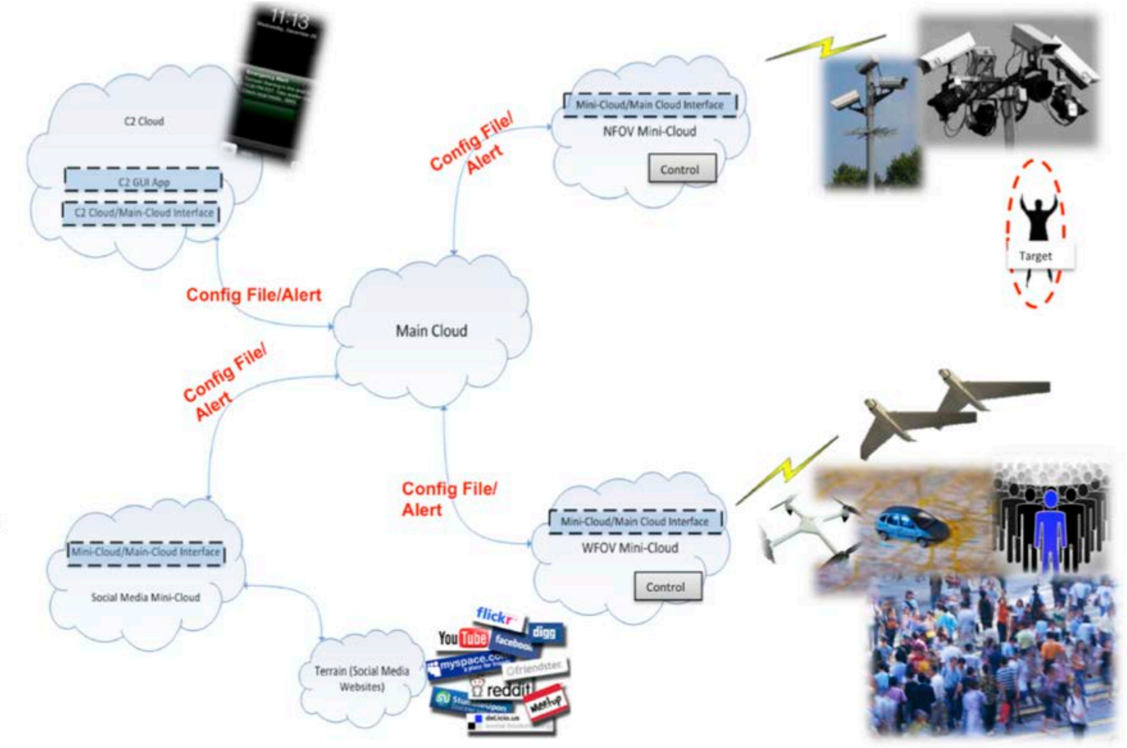


Figure 8. Application of smart cloud SoS during a humanitarian assistance and disaster relief operation (after Goshorn 2013).

From stakeholder opinion, there is concern that there are terrorist attacks in the area (both cyber and physical). For the smart cloud system of systems project, an architecture was put together detecting physical attacks in the area of the HA/DR operation.

For this thesis, a capability and architecture is proposed for how to detect cyber terrorist attacks within the smart cloud system of systems that is being used to perform surveillance for physical terrorist attacks.

E. CAPABILITY GAP AND PROPOSED CAPABILITY CONCEPT

This section first describes the capability gap of need for cyber situational awareness within a smart cloud system of systems, which this thesis addresses and is based on stakeholder input. This section then concludes by describing the concept of the proposed capability to fill the capability gap; that is, a cyber situational awareness system of systems.

1. Capability Gap

Whether it is a system used in real time during a mission or it is a system being assessed for vulnerabilities prior to being deployed, automation of cyber situational awareness (specifically information assurance) of any system supporting military operations is crucial. Operators in any mission need assurance that the data being used to support that mission is confidential, maintains its integrity, and is available throughout the mission. Additionally, before any system is deployed, such information assurance of that system needs to be categorized by the DoD in order to authorize its deployment (Young 2011; Onuskanich 2011).

Maintaining situational awareness of the data is even more crucial in a cloud environment where data is accessible to multiple users. Because data is more widely available to more users, it is well known that many governments are creating specific measures for how to assess information assurance in such cloud frameworks. One example of these measures is the United States' FedRamp organization (GSA 2014). Additionally, in cloud environments, the amount of data that needs to be monitored for vulnerabilities is overwhelming. There are not enough analysts to manually handle the data that are collected from such monitoring systems of a cloud framework (Hamel 2013). Thus, a systems approach to automate cyber situational awareness of data and systems within cloud systems and to augment analysts' work is needed.

Finally, in applications where the data collected will help save lives, as in natural disaster situations, it is important to ensure the data being collected has not been tampered with, but that it maintains its integrity (IDA 2013).

To fill these needs, this thesis chooses to apply an automated cyber situational awareness system to monitor social media data collected in a smart cloud system of systems. This will demonstrate an example of one type of data that needs to be monitored and for which its information confidentiality, integrity, and availability must be assured.

2. Proposed Concept

The proposed solution should provide the user with cyber situational awareness and command and control ability. There are two types of cyber defense: passive and active defense. The proposed solution should exploit these defenses and provide the ability to reinforce each other. This thesis will look at providing a high-level concept cyber situational awareness system of systems (SoS) with the ability to both passively monitor data and devices within a network, such as a smart cloud SoS, as well as actively monitor the network and devices within a smart cloud SoS.

Specifically, the concept will focus on passive defense to provide information assurance of data vulnerability within the smart cloud SoS by providing information assurance alerts with the usage of automating tools. Data collected within a smart cloud SoS vary from warfighting platforms such as weapon systems and sensor systems, and from main cloud stems. The potential loss of confidentiality, integrity, and availability of such data must be avoided, and thus there is a need to automate cyber situational awareness in a smart cloud SoS.

This thesis proposes a system of systems engineering methodology for integrating and configuring an intelligence automation system for the purpose of enabling cyber command and control (C2) and cyber situational awareness (SA). This methodology includes performing the systems engineering process on the overarching SoS with the main focus on just one of the supporting systems. In particular, this thesis first proposes an overarching cyber defense system of systems architecture demonstrating how systems work independently and together for enabling command and control and providing cyber situational awareness. It is first made up of the overarching Cyber SA and C2 (CSAC2) system, followed by two defense systems that are performing intelligence automation via the Detect-Identify-Predict-React framework (Figure 9). The two cyber defense systems are the Cyber Attack Detection SA (CADSA) system, which performs active cyber defense operations, and the Information Assurance (IA) SA (IASA) system, which performs passive cyber defense operations. This thesis documents the high-level system of systems architecture while focusing on the progress of the systems engineering process of the passive cyber defense system in particular.

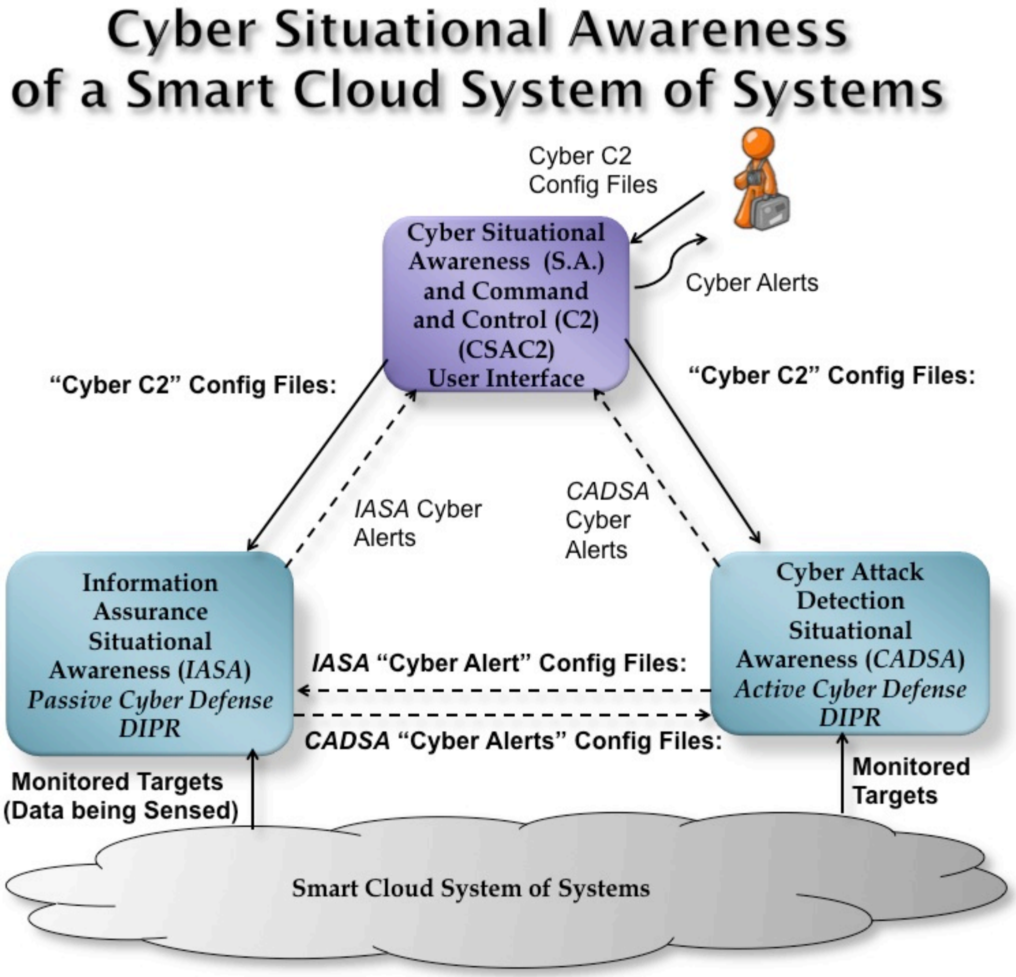

Cyber Situational Awareness of a Smart Cloud System of Systems
Cyber C2 Config Files
Cyber Situational Awareness (S.A.) and Command and Control (C2) (CSAC2) User Interface
Cyber Alerts
"Cyber C2" Config Files:
"Cyber C2" Config Files:
IASA Cyber Alerts
CADSA Cyber Alerts
Information Assurance Situational Awareness (IASA) Passive Cyber Defense DIPR
Cyber Attack Detection Situational Awareness (CADSA) Active Cyber Defense DIPR
IASA "Cyber Alert" Config Files:
CADSA "Cyber Alerts" Config Files:
Monitored Targets (Data being Sensed)
Monitored Targets
Smart Cloud System of Systems

Figure 9. Concept diagram for cyber situational awareness of a smart cloud system of systems. The cyber SA system of systems utilizes two parallel DIPR intelligence automation systems (one passive cyber defense and one active cyber defense) that both interact with each other and also independently provide cyber alerts to the overarching cyber SA system.

Thus, in this thesis, the systems engineering process is applied to the IASA system, which applies DIPR to perform passive cyber defense. After going through the systems engineering process and generating an architecture with an analysis of alternatives, this thesis implements and describes an instantiated proof-of-concept system of a portion of the IASA system, with documented test and evaluation results.

The initial conceptual requirements of this system of systems are documented as needing to be capable of (1) monitoring information assurance of information within specified systems in a cloud framework, along with (2) monitoring for specific cyber attacks on a specified system, (3) alerting information of interest to an operator, and finally (4) allowing the end user to perform command and control over his or her cyber assets.

3. Thesis Research Questions

In order to determine the apt solution for the IASA system, this section presents the primary and subsidiary research questions investigated during the thesis research for proposing a solution to the proposed concept.

Primary Research Question: How does one apply system of systems engineering to create an architecture for automating cyber situational awareness of specified information within a smart cloud system of systems?

Subsidiary Research Questions:

1. What functions would an automated cyber situational awareness system perform, in particular, one that automates information assurance of specified data?
2. How does one model and monitor cyber behaviors of specified information on a system?
3. What are the approaches to automate detecting and tracking degradation of information assurance (i.e., degradation of information confidentiality, integrity, and availability)?
4. What are the cyber sensors needed to capture data sufficient enough to automatically detect and track degradation of information assurance?

F. THESIS STRUCTURE

The need for automating information assurance for cyber situational awareness within the smart cloud system of systems was discussed in this chapter, followed by the proposed overall architecture of the systems. A general background and literature search on the controls to ensure confidentiality, integrity, and availability of data is described in Chapter II. In Chapter III, the detailed system diagrams of the proposed architecture are discussed with a focus on the IASA system using IDEF0 diagrams. The feature matrix used to demonstrate the concept of the Detect, Identify, Predict and React (DIPR) model for the IASA system is also mentioned. In Chapter IV, the proof of concept for the Detect and Identify of the DIPR model is demonstrated and the physical architecture of IASA is discussed. Finally, conclusions, recommended improvements to the proof of concept, and future research work are provided in Chapter V.

II. BACKGROUND

In addition to background on the systems engineering process used in this thesis, this chapter provides a brief overview of the three technologies used in this thesis, as well as background on the operational scenario selected for this thesis. The three technologies used are the smart cloud system of systems, Detect-Identify-Predict-React (DIPR), and information assurance. The operational scenario and assumptions in question are reviewed, followed by the brief overview of the systems engineering process used in this thesis.

A. SMART CLOUD SYSTEM OF SYSTEMS

1. Smart Cloud System of Systems

The smart cloud SoS is the assumed external system whose data needs are monitored for the cyber situational awareness system of systems. This infrastructure is made up of smart sensor mini clouds, a command and control mini cloud, and the main cloud, as shown in Figure 10. The command and control “mini clouds” represent nodes at tactical sites or at sites in which an operator and/or analyst is using data from the cloud to make a decision. The main cloud node represents a set of enterprise cloud nodes that perform further analysis of the data and is in charge of routing the right data to the right operator/analyst. The smart sensor mini cloud senses a particular domain, such as the physical domain, cyber domain, or social media domain, in order to generate raw sensor data that captures each domain at a certain time.

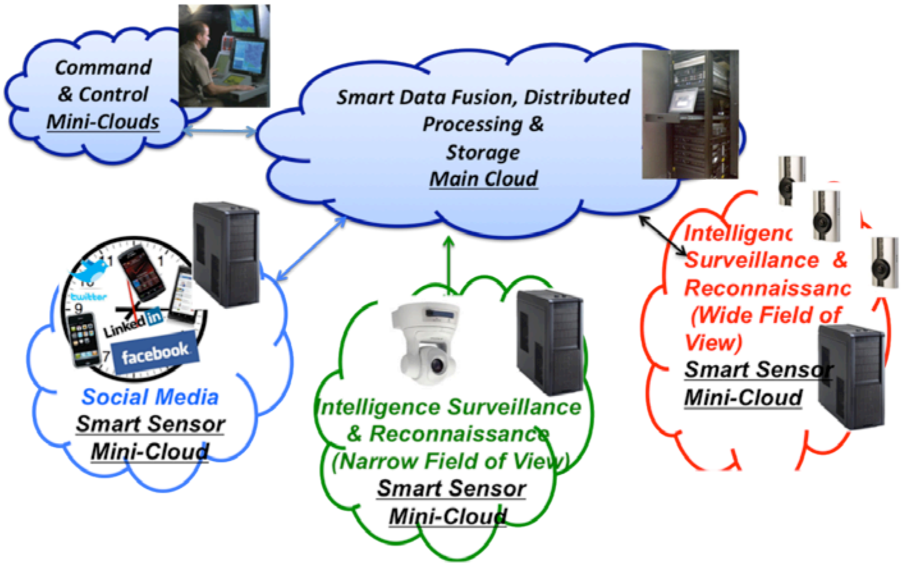


Figure 10. Smart cloud system of systems (after Goshorn 2013).

In other words, the smart sensor mini clouds collect and analyze data from social media and video feed data from the Intelligence Surveillance and Reconnaissance (ISR) system, and provide output in a standardized alert data file format to the main cloud. The main cloud further fuses these data, distributes it, and stores the data on the main cloud. The command and control mini cloud serves as the hub to gather input from the operators and also to provide cyber situational awareness in the form of alerts. It is important to monitor the IA of the data in its different states of data at rest, in transit, and in use within the smart cloud system of systems.

B. DETECT, IDENTIFY, PREDICT, REACT INTELLIGENCE AUTOMATION MODEL

The intelligence automation model, Detect-Identify-Predict-React, or DIPR, has been used primarily in sensor-based networks to generate usable, intelligent feedback from raw data provided by conventional sensors such as cameras (Goshorn 2011). It has been used also for automating dynamic defensive cyber operations, which focused primarily on the cyber sensors and detection portions (Jurjonas 2012; Deptula 2013).

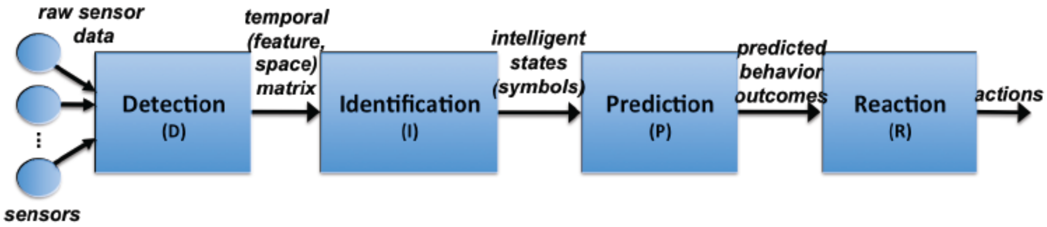

raw sensor data
sensors
Detection (D)
temporal (feature, space) matrix
Identification (I)
intelligent states (symbols)
Prediction (P)
predicted behavior outcomes
Reaction (R)
actions

Figure 11. Concept of DIPR model and its subsystem (after Goshorn 2011).

In addition to the sensor systems, there are four intelligence automation systems that further process the sensor data, providing alerts that are increasing in complexity at each output of each system. Therefore the alerts that are output from Reaction are the most complex where the sensor data is the lowest level of information, in terms of complexity.

This section briefly discusses each DIPR subsystem: Detect, Identify, Predict, and React.

1. Detect System

Sensors deployed in smart sensor mini clouds provide raw data for processing by a system/processor running the Detect programs. The raw sensor data is analyzed by the Detect subsystem, which creates the detect classifications in the data feature matrix. Features are a low-level classification created from raw data and are describing an object of interest (Goshorn 2011).

2. Identify System

The Identify subsystem processes the Detect features, recognizes when multiple features or conditions have occurred simultaneously, and fuses the data together, generating an intelligent state to which that data belongs. Rules to implement the recognition must be developed and input into the program to determine what feature conditions are required. Ideally, it would be an adaptable learning system that could tailor itself over time (Goshorn 2011).

3. Predict System

At each time interval, the intelligent states generated by the Identify subsystem are then input into the Predict subsystem. These states are then processed by a generalized high-level classifier that recognizes spatiotemporal patterns of states. The input states that also include geo-locations and time of events associated with each state form object sequences, or behaviors. They are then classified as "normal" behaviors, "abnormal" behaviors, and "unclassified" behaviors, based on predefined or learned patterns. These behaviors are then used to predict the future state of the information under observation and output to the React subsystem (Goshorn 2011).

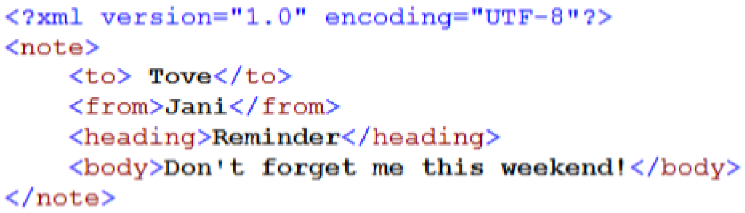
4. React System

The React subsystem provides an appropriate action (control signal and/or alert) based on these predefined rules of engagement associated with each predicted behavior (Goshorn 2011). For example, a control signal to close network ports or encrypt files or folders may be issued as a form of reaction to predict behaviors.

5. Extensible Markup Language (XML) Interoperable File Standard for the DIPR Alerts

Each object that is detected in the Detect stage within DIPR has its own feature space matrix, which is a data structure that stores all the values of the object features that were detected. There are several methods to implement the feature space matrix. One common method is to use the Extensible Markup Language (XML) file standard (w3school 2014).

XML is a structured language, such as HyperText Markup Language (HTML), which was created to structure, transport, and store data. The structure of how data is organized in the document is customized by whoever is creating the XML files (w3school 2014). Figure 12 shows a sample of an XML document.



```
<?xml version="1.0" encoding="UTF-8"?>
<note>
    <to> Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

Figure 12. XML document example (from w3schools 2014).

The XML standard is well suited for DIPR application to the cyber field. In Chapter III, more specific details of the adaptation are described to show how the subsystems and functions work to provide automated information assurance and situational awareness for a smart cloud system of systems.

C. SOCIAL MEDIA AND HUMAN ASSISTANCE/DISASTER RELIEF OPERATIONS

Recall from Chapter I that the operational scenario selected describes the deploying of a cyber situational awareness system of systems to monitor for cyber attacks against the smart cloud system of systems that is monitoring for physical terrorist attacks in a region in Singapore that had hypothetically been hit with a natural disaster, all as a part of an HA/DR operation.

This scenario also assumes that social media is used by people affected by the disaster during the HA/DR operations. Additionally, this selected HA/DR scenario of a smart cloud system of systems within Singapore assumes that social media is used in Singapore, since one of the smart sensor mini clouds is a social media smart sensor mini cloud that could sense all publicly available social media generated within Singapore.

To ensure that this deployment of a smart cloud system of systems is valid, and that its data is being generated, would require situational awareness of its information assurance; this section describes both the background on the social media usage within Singapore as well as the background on social media usage during HA/DR operations.

Finally, to provide background on what sample social media looks like, this section finishes with a brief example of what Twitter "raw sensor data" looks like. For this thesis, this is the type of data for which information assurance would be monitored during an HA/DR scenario.

1. Social Media in Singapore

According to a report by Rock Publicity in 2012, social media is heavily adopted in Singapore. In particular, 68.1% of the total population of 3,530,100 in 2012 regularly interacted with social media on their devices at home or anywhere on their mobile devices (Rock Publicity 2012).

With respect to which social media applications are used, most Singaporeans have accounts with multiple social media sites. In fact, Singaporeans are some of the heaviest mobile users on average in Asia, and use social media on mobile devices more than 96% of the rest of the world. Interestingly, they also buy more tablet devices per capita than all of the other Asian nations, appearing to lead the Asian region with respect to the most social media used on these devices. Additionally, almost half of the country's total population is reported to have a Twitter account, which is higher than the world average. In terms of which city in the world uses Twitter the most, Singapore is reported to be ranked eleventh. In particular, it is reported that the average Singaporean Twitter user tweets more than twice per day (Rock Publicity 2012).

To conclude, it is reasonable to consider that social media data generated within Singapore will be a valid form of sensor data requiring information assurance to be monitored within the smart cloud system of systems.

2. Use of Social Media in HA/DR Operations

This section reviews the usage of social media applications during HA/DR operations.

In general, social media is used to maintain connections with family and friends, with some used to learn more about consumer products, interact with those who enjoy a common hobby or interest, and also to meet new acquaintances. Additionally, there are many reasons why people use social media during disasters. During a disaster scenario, people may use social media to look for both readily available information and information that has a lot of content (Fraustino et al. 2012).

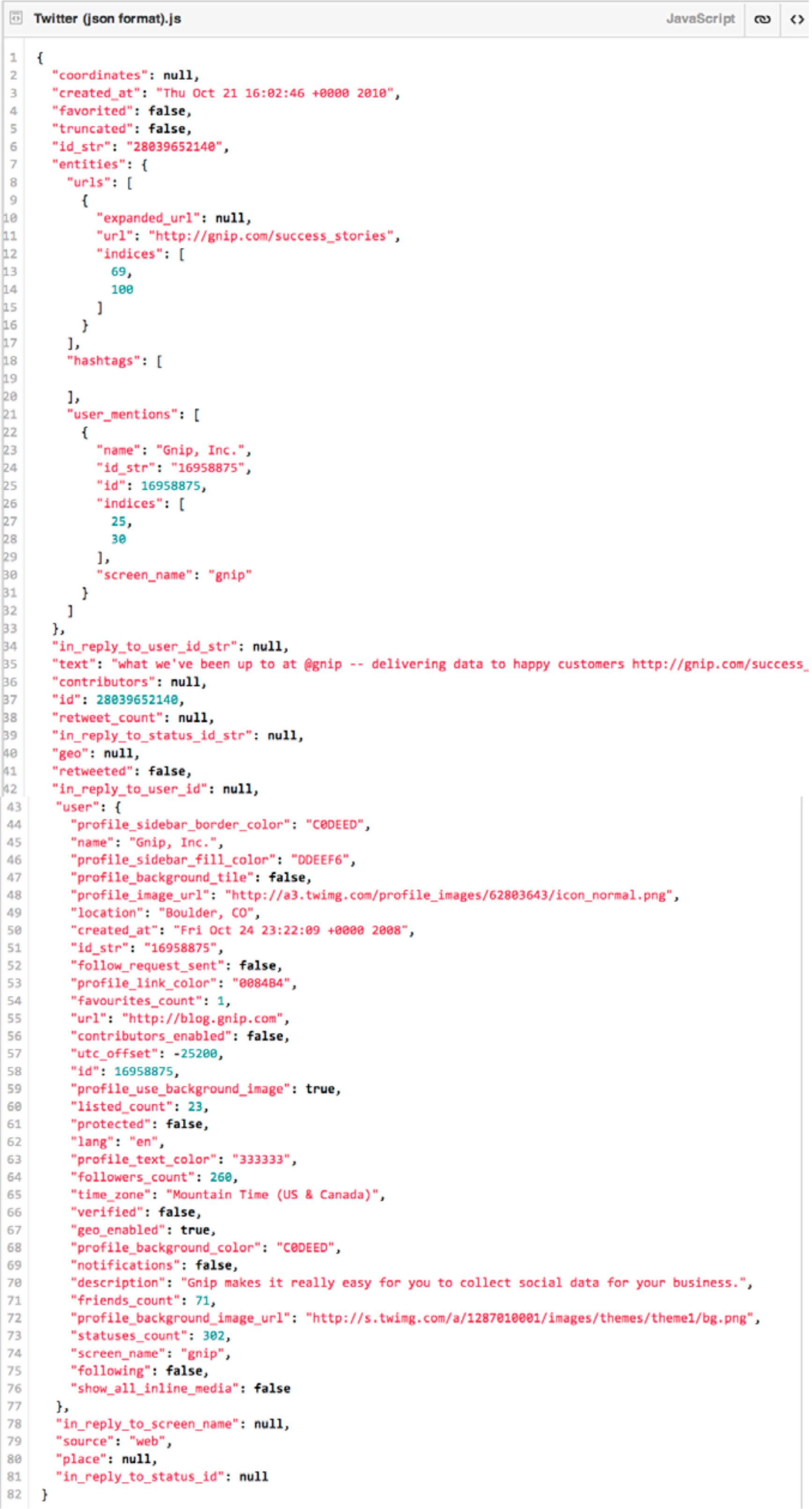
There are many examples of when people have used social media during disaster situations. First, half an hour before a potentially fatal storm hit a festival in Belgium in 2011, it is reported that people tweeted more than 2,000 related tweets. Then, after the first four hours had passed in this disaster, this number increased to more than 80,000 tweets (Perng et al. 2012). Another example of using social media during a disaster occurred during the 2008 earthquake in China. Interestingly, in this case, the first reports describing the disaster did not come from the government, but rather from Twitter (Mills et al. 2009). The sources of several of the tweets were from both local and national news media. In addition to the media, an unexpected number of tweets were created by Twitter account users that were specific to disasters, as well as ordinary citizens that reported tweets for the purpose of updating other citizens with helpful information (Mims 2010).

3. Social Media Sensor Data

This section provides a brief example of what Twitter "raw sensor data" looks like. For this thesis, this is the type of data for which information assurance would be monitored during an HA/DR operation.

A social media sensor can be thought of as a software program that senses a specific social media platform (Goshorn 2013). For example, the Twitter Application Programming Interface (API) is an application programming interface that allows anyone with a Twitter account to automatically sense a small percentage of all public tweets in real time (for streaming) or in non-real time, which searches a subset of all tweets within a specified timeframe (batch processing). The Twitter API is made up of two discrete APIs: the Representational State Transfer (REST) API and a Streaming API. It presently supports the following data formats: XML, JSON, and the RSS and Atom syndication formats (Twitter Developers 2012).

A sample of the Twitter sensor data is illustrated in Figure 13. The current format used is JavaScript Object Notation (JSON format).



```
{
  "coordinates": null,
  "created_at": "Thu Oct 21 16:02:46 +0000 2010",
  "favorited": false,
  "truncated": false,
  "id_str": "28039652140",
  "entities": {
    "urls": [
      {
        "expanded_url": null,
        "url": "http://gnip.com/success_stories",
        "indices": [
          69,
          100
        ]
      }
    ],
    "hashtags": [

    ],
    "user_mentions": [
      {
        "name": "Gnip, Inc.",
        "id_str": "16958875",
        "id": 16958875,
        "indices": [
          25,
          30
        ],
        "screen_name": "gnip"
      }
    ]
  },
  "in_reply_to_user_id_str": null,
  "text": "what we've been up to at @gnip -- delivering data to happy customers http://gnip.com/success_
  "contributors": null,
  "id": 28039652140,
  "retweet_count": null,
  "in_reply_to_status_id_str": null,
  "geo": null,
  "retweeted": false,
  "in_reply_to_user_id": null,
  "user": {
    "profile_sidebar_border_color": "C0DEED",
    "name": "Gnip, Inc.",
    "profile_sidebar_fill_color": "DDEEF6",
    "profile_background_tile": false,
    "profile_image_url": "http://a3.twimg.com/profile_images/62803643/icon_normal.png",
    "location": "Boulder, CO",
    "created_at": "Fri Oct 24 23:22:09 +0000 2008",
    "id_str": "16958875",
    "follow_request_sent": false,
    "profile_link_color": "0084B4",
    "favourites_count": 1,
    "url": "http://blog.gnip.com",
    "contributors_enabled": false,
    "utc_offset": -25200,
    "id": 16958875,
    "profile_use_background_image": true,
    "listed_count": 23,
    "protected": false,
    "lang": "en",
    "profile_text_color": "333333",
    "followers_count": 260,
    "time_zone": "Mountain Time (US & Canada)",
    "verified": false,
    "geo_enabled": true,
    "profile_background_color": "C0DEED",
    "notifications": false,
    "description": "Gnip makes it really easy for you to collect social data for your business.",
    "friends_count": 71,
    "profile_background_image_url": "http://s.twimg.com/a/1287010001/images/themes/theme1/bg.png",
    "statuses_count": 302,
    "screen_name": "gnip",
    "following": false,
    "show_all_inline_media": false
  },
  "in_reply_to_screen_name": null,
  "source": "web",
  "place": null,
  "in_reply_to_status_id": null
}
```

Figure 13. Twitter sample payload, JSON format (from GitHub 2014).

D. INFORMATION ASSURANCE

This section covers the characteristics of data that determines information assurance, the three different states that data can be in, the 20 critical security controls, and the type of cyber sensors that can be used to achieve the goal of low-level automation/feature extraction from sensor data describing information assurance of specified data.

1. Confidentiality, Integrity, and Availability

Information assurance can be determined by three characteristics of data: *confidentiality*, *integrity*, or *availability*. Confidentiality of data means that unauthorized users cannot access the data. Integrity of data means data unauthorized users cannot change the data. Availability of data means that data is readily available to authorized users (UM School of Medicine 2006).

2. Data at Rest, Data in Transit, and Data in Use

Data can exist in any one of three states, as illustrated in Figure 14 (Ball 2013). The states are “data at rest,” “data in transit,” and “data in use.”

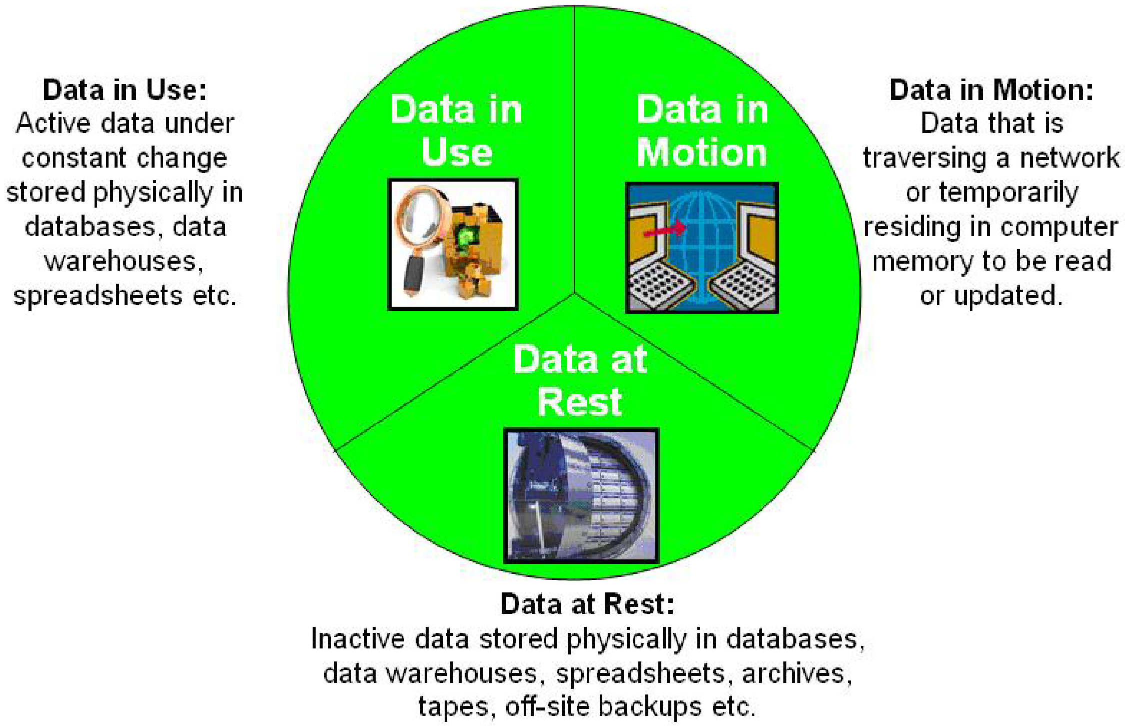

Data in Use:
Active data under constant change stored physically in databases, data warehouses, spreadsheets etc.
Data in Use
Data in Motion
Data in Motion:
Data that is traversing a network or temporarily residing in computer memory to be read or updated.
Data at Rest
Data at Rest:
Inactive data stored physically in databases, data warehouses, spreadsheets, archives, tapes, off-site backups etc.

Figure 14. The three states of data are "data in use," "data at rest," and "data in motion" (from Ball 2013).

Data Loss Prevention (DLP) is one type of methodology to identify, monitor, and protect information. There are many ways to do this, such as deep inspection, contextual security analysis of transactions, and finally through a central management framework. The first purpose of the DLP systems is to detect unauthorized use of information that should remain confidential. The second purpose is to prevent unauthorized use of information that should remain confidential. Finally, the last purpose is to prevent unauthorized transmission of the data (Norton 2011).

For data at rest, DLP can be achieved by locating and cataloging sensitive information stored. While monitoring and controlling the movement of sensitive information across the enterprise network is essential for data in motion/transit, the same can be said for monitoring and controlling the movement of sensitive information on end-user systems for data in use or in a process.

a. Data at Rest (Stored Data)

Data at rest refers to data that is not in used and is stored on a storage device such as a hard disk, CD/DVD, or flash memory device in the form of files or database. Data protection for the data at rest state can be in the form of data encryption and access logging.

b. Data in Motion/Transit (Transmit Data)

Data in motion or transit refers to data that is being transmitted or moving between applications (within the computer) or networks (within a network) using available communication linkage such as a local area network (LAN), Wifi, or computer bus. Data protection for data in transit can be in the form of constant monitoring and protection of data such as Transport Layer Security (TLS), Secure Sockets Layer (SSL), and Internet Protocol Security (IPsec) protocols, and usage of data loss prevention software. Protocols such as TLS, SSL, and IPsec encrypt data packets for secure transportation and decryption by authorized parties. Data loss prevention software monitors network traffic to help prevent unauthorized transmission of data within and exiting enterprise network. In addition, there should be a policy implemented to prevent a user's negligence in data handling.

c. Data in Use/Process

Data in use are data that are actively being read, modified, or managed by an application, and the data is stored temporarily in memory, such as random-access memory (RAM) or Page Files. This is the most vulnerable state as the data must allow changes to be made and are not protected from attack; therefore, the data requires constant monitoring. Data protection for data in use can be achieved by ensuring access only by trusted applications or authorized users, preventing snooping by third-party applications, and employing full memory encryption.

3. Cyber Sensors for Information Assurance

This section provides a brief background on cyber sensors in general, based out of previous research (Goshorn 2009).

There are two ways of deploying cyber sensors in general. They are classified by where are the monitoring is required—that is, monitoring of the network or monitoring of network devices, also known as the host. Therefore, cyber sensors can be generally categorized by the location in which they have been deployed, either network based or host based (Deptula 2013).

Specific to automating information assurance, network-based cyber sensors are used to monitor information in the state of data in transit, while host-based cyber sensors are used to monitor information in the state of data at rest and data in use.

Furthermore, automation of information assurance requires the data to be monitored for loss of confidentiality, integrity, and availability. Therefore, when choosing sensors, not only is the data state important, the ability to capture the probabilities of the data maintaining confidentiality, integrity, and availability, based on extracted features of the sensor data, is important. This concept is further discussed in Chapter III.

E. CRITICAL SECURITY CONTROLS

The SANS Institute worked with information assurance and cybersecurity experts from government and industry to determine the highest priority controls that one could employ in order to have an effective cyber defense system. For example, information assurance requires data's confidentiality, integrity, or availability is preserved. Recommendations put forward by the SANS Institute's Critical Control 17 describes the application of Data Loss Prevention (DLP) in order to actively monitor both data at rest and data in transit (SANS Institute 2014). This control provides the initial concept for achieving information assurance of the data.

Out of the 20 controls they identified, eight critical controls are found to be the most applicable to automating information assurance in the smart cloud system of systems. The eight controls used in this thesis are described subsequently, with the most relevant control being Control 17 (SANS Institute 2014).

1. Critical Control 1: Inventory of Authorized and Unauthorized Devices

Control 1 requires that an organization detects and maintains an up-to-date inventory of both authorized and unauthorized devices within the network (SANS Institute 2014). This concept is depicted in Figure 15.

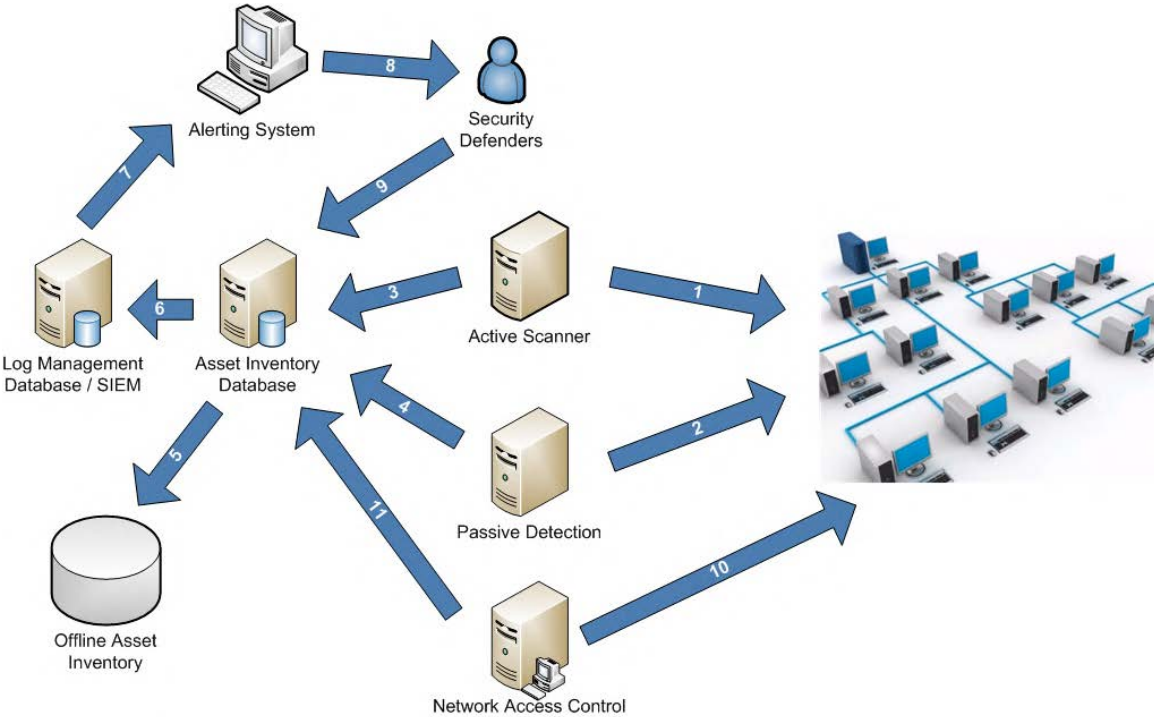


Figure 15. Control 1 system entity relationship diagram (from SANS n.d.).

2. Critical Control 3: Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers

Control 3 requires that organizations ensure their hardware and software for mobile devices, laptops, workstations, and servers are configured for security (SANS Institute 2014). This is depicted Figure 16.

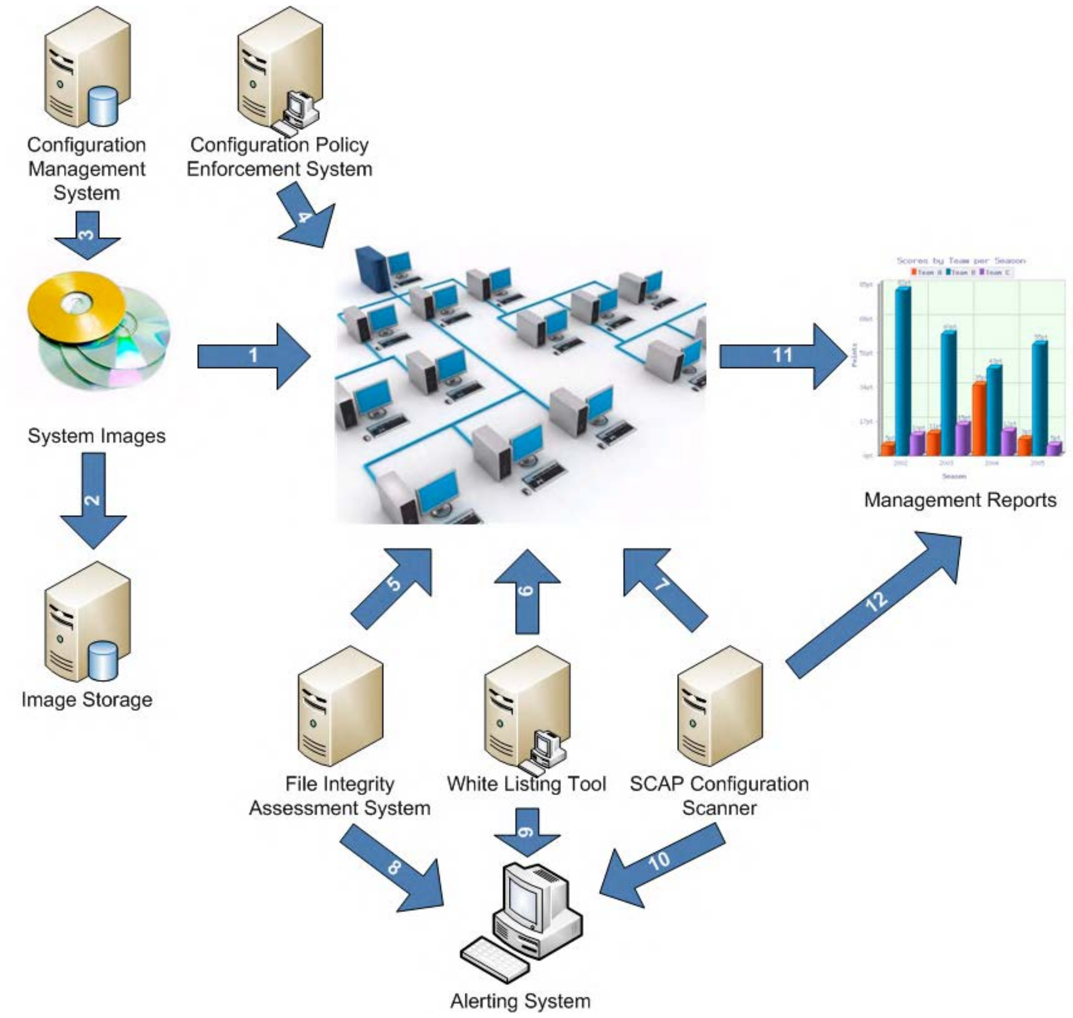
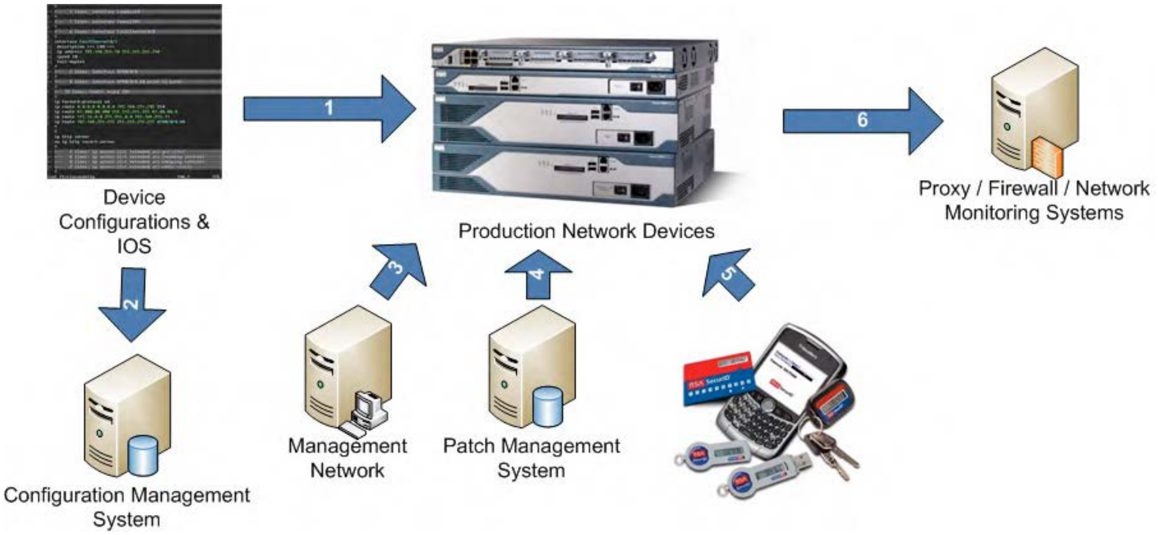

Configuration Management System
Configuration Policy Enforcement System
3
4
System Images
1
11
Scores by Team per Season
Management Reports
2
5
6
7
12
Image Storage
File Integrity Assessment System
White Listing Tool
SCAP Configuration Scanner
8
9
10
Alerting System

Figure 16. Control 3 system entity relationship diagram (from SANS Institute 2014).

3. Critical Control 10: Secure Configurations for Network Devices

Control 10 requires that all network devices (such as routers, firewalls, switches) are configured for security, as depicted in Figure 17 (SANS Institute 2014).



Device Configurations & IOS
1
Production Network Devices
6
Proxy / Firewall / Network Monitoring Systems
2
3
4
5
Configuration Management System
Management Network
Patch Management System

Figure 17. Control 10 system entity relationship diagram (from SANS Institute 2014).

4. Critical Control 11: Limitation and Control of Network Ports, Protocols, and Services

Control 11 requires that network ports, protocols, and services within each computer system on a network be limited to allow input and output of required data doing required services (SANS Institute 2014). This concept is depicted in Figure 18.

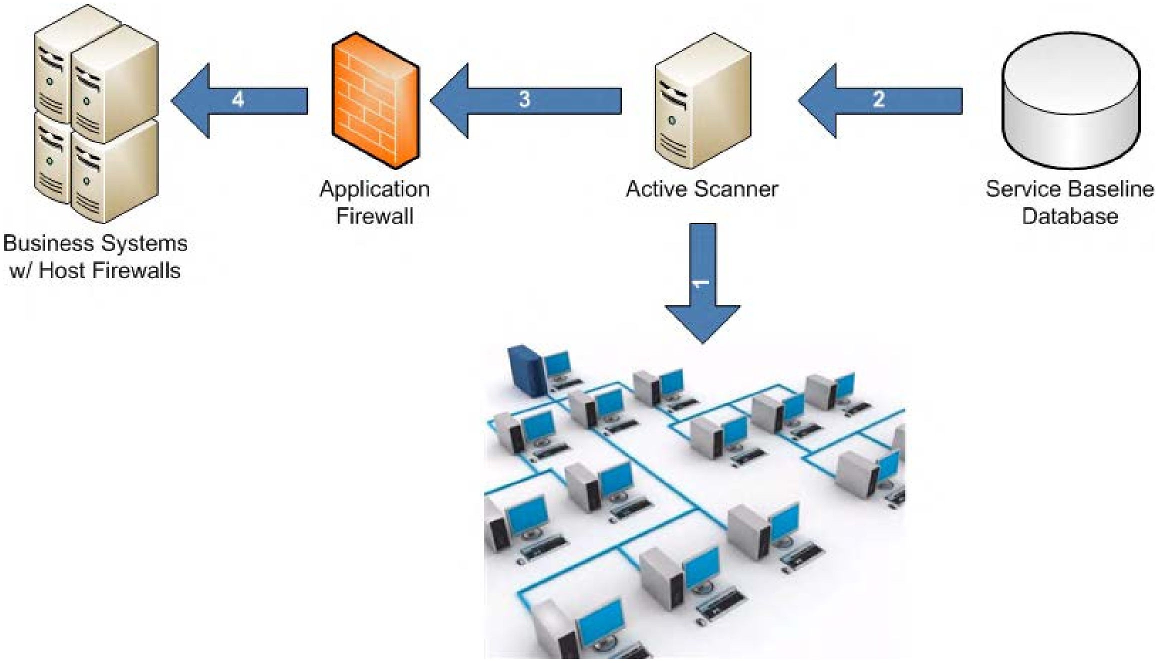


Figure 18. Control 11 system entity relationship diagram (from SANS Institute 2014).

5. Critical Control 13: Boundary Defense

Control 13 requires that a network have layers of defense by employing protective/monitoring systems throughout the network to create hierarchical boundaries in the network (SANS Institute 2014). This concept is depicted in Figure 19.

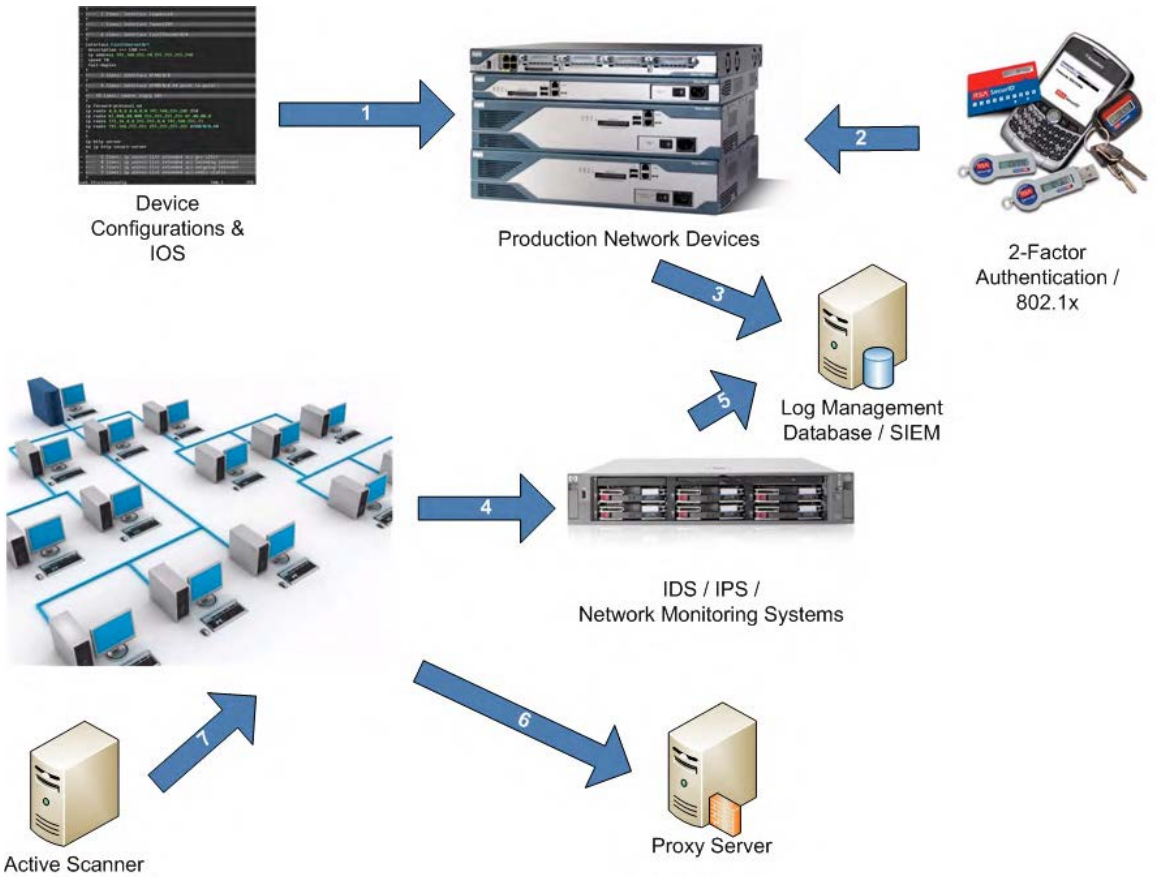

Device Configurations & IOS
1
Production Network Devices
2
2-Factor Authentication / 802.1x
3
Log Management Database / SIEM
5
4
IDS / IPS / Network Monitoring Systems
7
6
Active Scanner
Proxy Server

Figure 19. Control 13 system entity relationship diagram (from SANS Institute 2014).

6. Critical Control 14: Maintenance, Monitoring, and Analysis of Audit Logs

Control 14 requires that all organizations maintain, monitor, and actively analyze logs that audit both the network and the systems on the network (SANS Institute 2014). This concept is depicted in Figure 20.

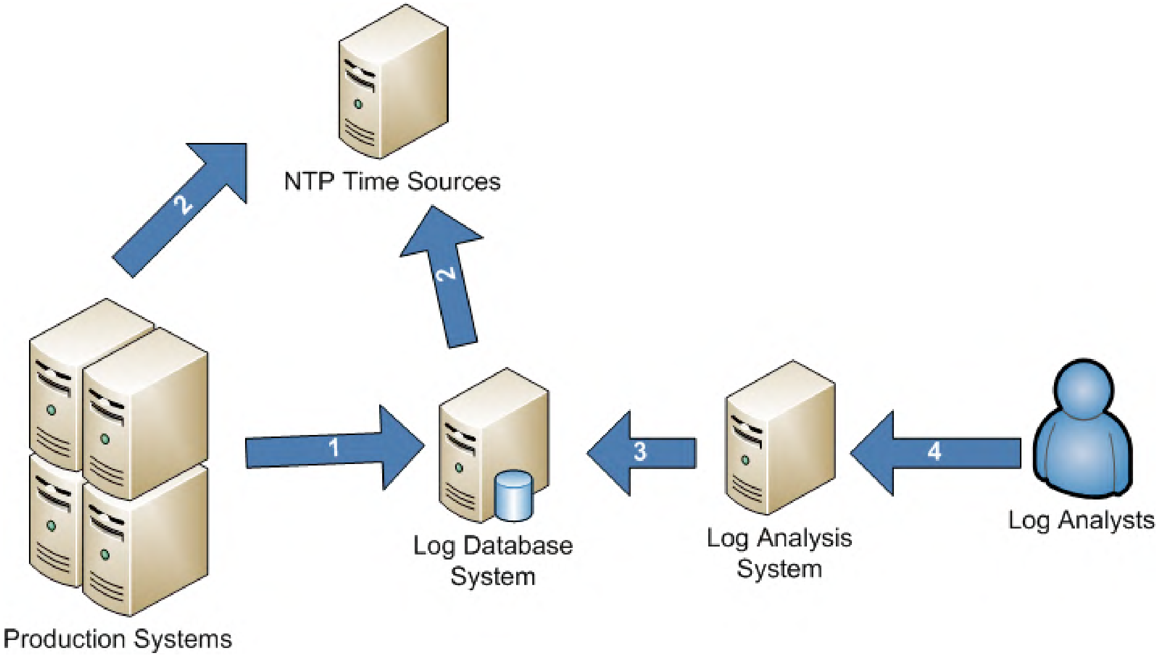


Figure 20. Control 14 system entity relationship diagram (from SANS Institute 2014).

7. Critical Control 17: Data Loss Prevention

Control 17 requires that the network be actively monitoring for data loss by employing scanning devices on data storage systems as well as on the network (SANS Institute 2014). This concept is depicted in Figure 21.

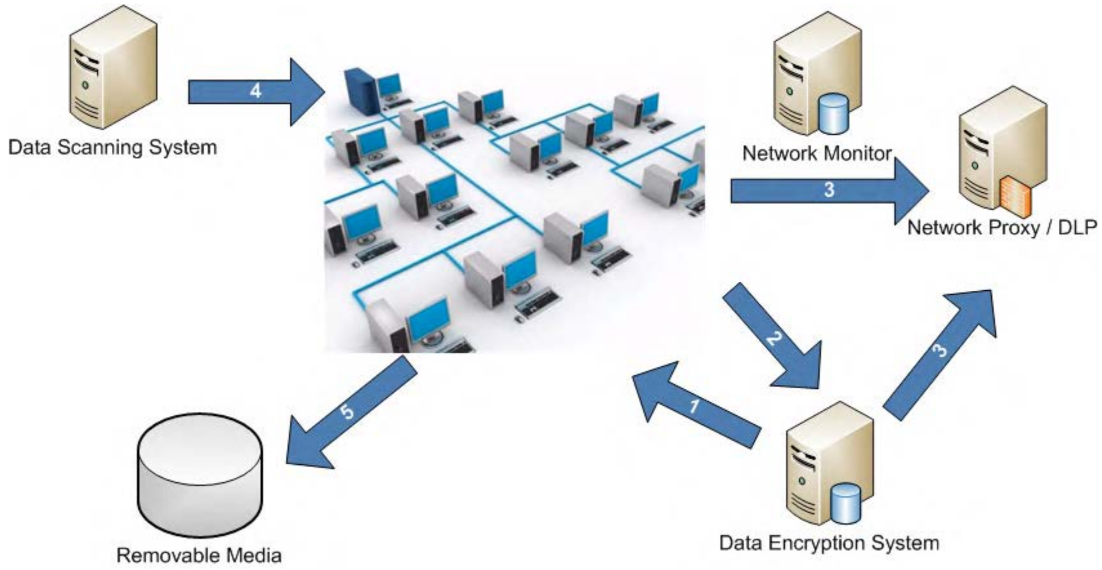


Figure 21. Control 17 system entity relationship diagram (from SANS Institute).

8. Critical Control 19: Secure Network Engineering

Control 19 requires that the engineering that goes into architecting and implementing an organization's network should be inherently secure in its design (SANS Institute 2014). This concept is depicted in Figure 22.

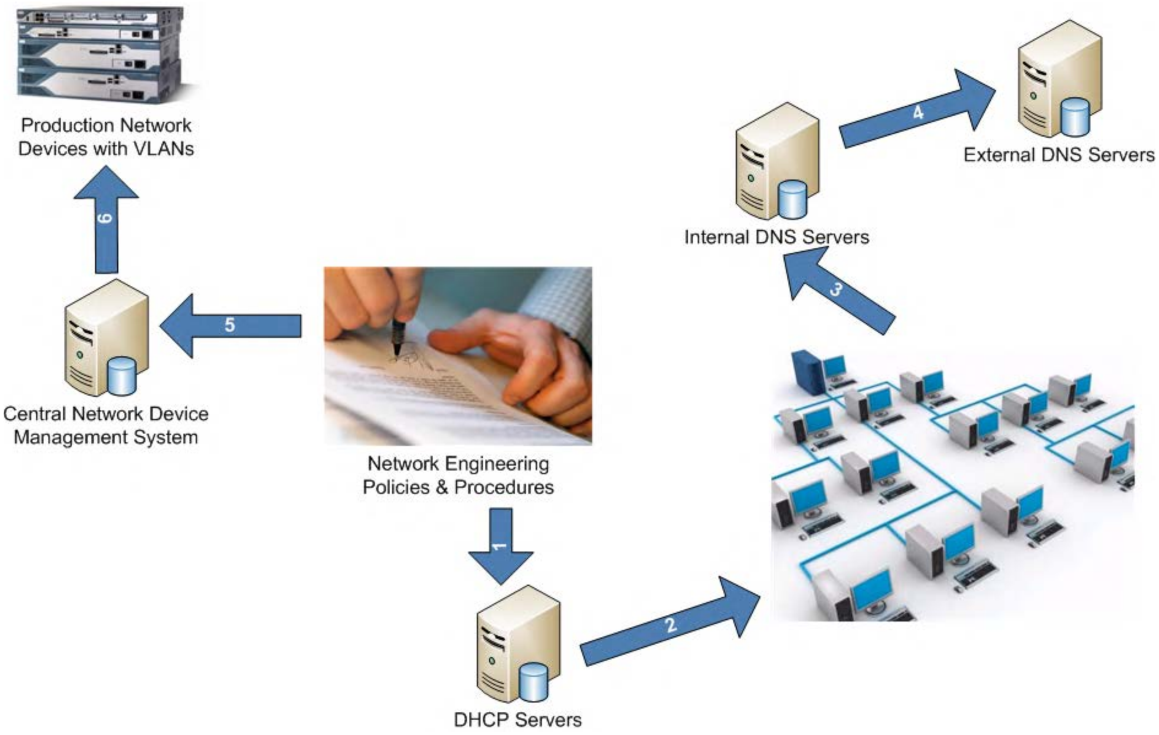


Figure 22. Control 19 system entity relationship diagram (from SANS Institute).

F. SYSTEMS ENGINEERING PROCESS

This section gives a brief review of the systems engineering model, the Vee model, used in architecting the new capability presented in this thesis (Buede 2009).

The Vee model represents the lifecycle development of large scale systems and software development projects, with an emphasis on the engineering process for a system being developed. The left side of the Vee represents the beginning three phases of life cycle development. The process starts from the top left of the Vee with the definition of the stakeholders' operational need; the operational requirements are decomposed, as we move from left to right in Figure 23, to system level requirements to specifications of each component. The horizontal line that is drawn under the middle intersection of the Vee represents the start of the design process of the product by the discipline engineers. Examples of such are electrical engineers, mechanical engineers, chemical engineers, civil engineers, aerospace engineers, and computer science engineers. The horizontal line also represents the part of the process in which there is overlap between the design and integration activities. The right side of the Vee indicates the integration and qualification activities of the system engineering. It involves putting together the lower-level components into higher level components and finally the assembly of high-level components into the system. All the activities up the Vee involve validating and verifying the newly assembled system elements to determine that each element meets the requirements or specifications that were established in the design phase and that the system meets the stakeholders' needs (Buede 2009).

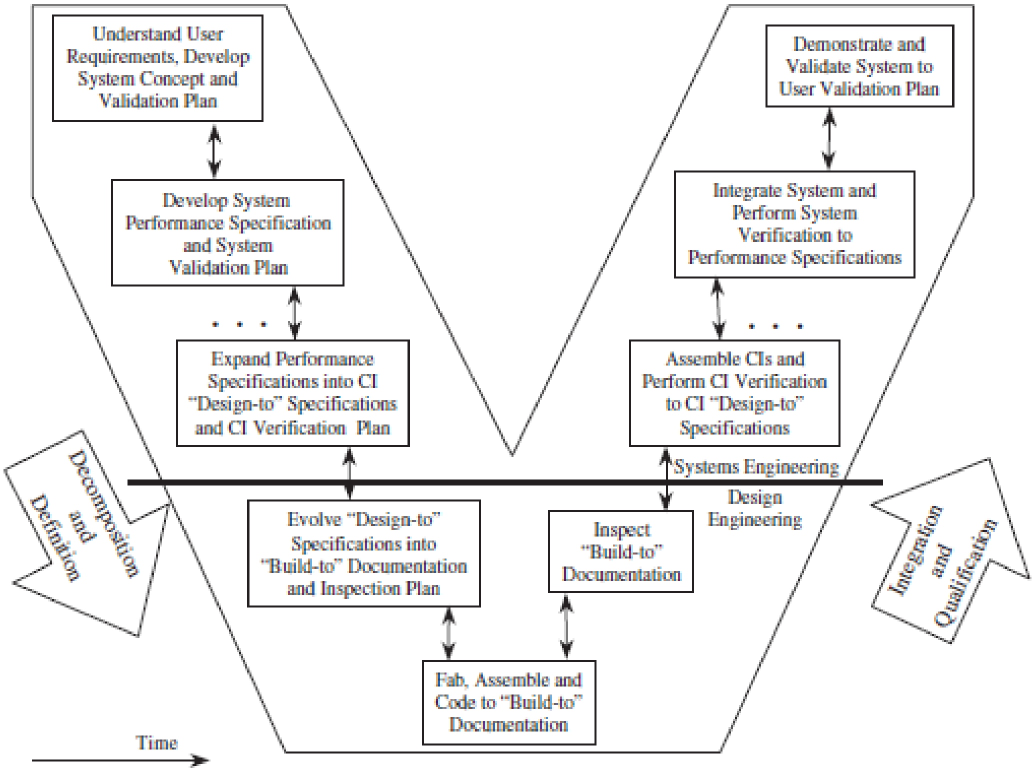

Understand User Requirements, Develop System Concept and Validation Plan
Develop System Performance Specification and System Validation Plan
Expand Performance Specifications into CI "Design-to" Specifications and CI Verification Plan
Evolve "Design-to" Specifications into "Build-to" Documentation and Inspection Plan
Fab, Assemble and Code to "Build-to" Documentation
Inspect "Build-to" Documentation
Assemble CIs and Perform CI Verification to CI "Design-to" Specifications
Integrate System and Perform System Verification to Performance Specifications
Demonstrate and Validate System to User Validation Plan
Systems Engineering
Design Engineering
Decomposition and Definition
Integration and Qualification
Time

Figure 23. Systems engineering "Vee" model (from Buede 2009, p. 10).

THIS PAGE INTENTIONALLY LEFT BLANK

III. PROPOSED ARCHITECTURE

Recall that in Chapter I we reviewed products generated during the conceptualization phase of the solution presented in this thesis. This is the first phase of the systems engineering Vee model, as introduced in Chapter II. This chapter reviews products implemented in the following phases of the Vee Model. Namely, the first section reviews the operational scenarios analysis to develop the system performance and requirements. Next, the functional definition and decomposition is conducted to determine the required functions of the system. The IDEF0 function modeling method is used to model the system data flow and control of the proposed IASA system. Requirements analysis will provide the starting point for tracing of the physical solution to the system requirements. Mapping of the physical solution to the functions of the IASA is shown in the form of traceability matrix. The validation and verification of a portion of the proposed system will be discussed in Chapter IV in the proof-of-concept section. As this thesis is focused on the functions of IASA, the physical architecture will similarly focus on the physical architecture of IASA only, as highlighted in Figure 24.

A. CYBER SITUATIONAL AWARENESS AND COMMAND AND CONTROL SYSTEM OF SYSTEMS CONCEPT

This thesis proposes a system of systems engineering methodology for integrating and configuring an intelligence automation system for the purpose of enabling cyber command and control (C2) and cyber situational awareness (SA). This methodology includes performing the systems engineering process to at least one of the supporting systems. To approach this topic, this thesis first proposes an overarching system of systems architecture demonstrating how systems work independently and together for enabling command and control and providing cyber situational awareness. This system of systems is depicted in Figure 24. It is first made up of the overarching Cyber SA and C2 (CSAC2) system, followed by the Cyber Attack Detection SA (CADSA) system, and finally, the Information Assurance (IA) SA (IASA) system. This thesis will document the high-level system of systems architecture while focusing on the progress of the systems engineering process of one system in particular. In this thesis, the systems engineering process is applied to the IASA system. After going through the systems engineering process and generating an architecture with an analysis of alternatives, this thesis implements and describes an instantiated proof-of-concept system of a portion of the IASA system, with documented test and evaluation results.

The initial requirements of this system of systems are documented as needing to be capable of (1) monitoring information assurance of information within specified systems in a cloud framework, along with (2) monitoring for specific cyber attacks on a specified system, (3) alerting information of interest to an operator, and finally (4) allowing the end user to perform command and control over his or her cyber assets.

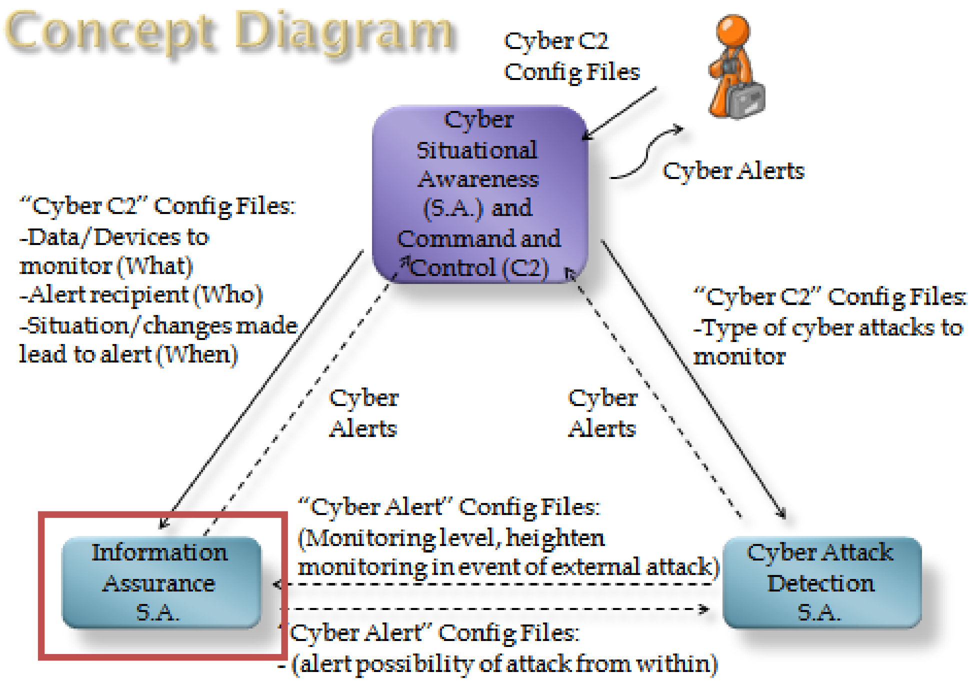

Concept Diagram
Cyber C2 Config Files
Cyber Situational Awareness (S.A.) and Command and Control (C2)
Cyber Alerts
"Cyber C2" Config Files:
-Data/Devices to monitor (What)
-Alert recipient (Who)
-Situation/changes made lead to alert (When)
"Cyber C2" Config Files:
-Type of cyber attacks to monitor
Cyber Alerts
Cyber Alerts
"Cyber Alert" Config Files:
(Monitoring level, heighten monitoring in event of external attack)
Information Assurance S.A.
Cyber Attack Detection S.A.
"Cyber Alert" Config Files:
(alert possibility of attack from within)

Figure 24. Focus on IASA in the cyber situational awareness system of systems of specified information in the cloud (boxed in red).

1. Detailed Discussion on Preliminary Proposed System of Systems Architecture

First, there is the Cyber SA system that interfaces with the operator. In order to obtain cyber situational awareness, an operator should be able to subscribe to an automated cyber situational awareness cloud service of interest, where he or she determines both the systems and information to be monitored (performing command and control over all cyber monitoring sensors/systems associated with that cyber situational awareness mission). Such a system also provides a visualization of the situational awareness of the desired systems, including alerts of interest.

Cyber situational awareness defense can be defined and decomposed into two main areas: passive cyber defense, such as information assurance, formerly termed by the U.S. Cyber Command as Department of Defense (DoD) GIG Operations (DGO), along with active cyber defense, such as cyber attack detection, termed DoD Cyber Operations (DCO) (Young 2011). Since cyber defense is decomposed into these two types of operations, the cyber situational awareness system needs two different monitoring systems: Active Cyber Defense SA systems and Passive Cyber Defense (Information Assurance) SA systems. Figure 24 represents the latter two systems denoted as CADSA system and IASA system. Both systems are proposed to include cyber sensors and intelligence automation systems in order to automate situational awareness of their cyber terrain. These systems are configured by the overarching cyber situational awareness system, to which they both provide alerts. Additionally, it is proposed that the two supporting systems provide alerts to each other. Thus, these three systems (the overarching Cyber SA system, the IASA system, and the CADSA system) interact with each other in a system of systems (SoS) environment, as shown in Figure 24. The benefit of having these applications in an SoS architecture is that it ensures synergy between the applications and produces a robust setup. This setup enables each system to support the others to achieve the desired results of assuring information confidentiality, integrity, and availability. Finally, the system of systems works together in alerting the operators in the event of cyber attacks or degradation in information assurance.

a. Cyber Situational Awareness (Cyber SA) System

The overarching Cyber SA System is the C2 system that acts as a gateway between the cloud architecture and the human operator. The cyber behaviors to be monitored are determined by the operators and uploaded here. Monitoring parameters can be in the form of the types of files to monitor and on what devices, along with certain specified behaviors such as change in a file's properties, and cyber attacks. In such events the alerts are sent to the operators and the recipients of the alerts. The type of data to be monitored, which is sometimes called a "datacube," is also specified here. Usually this datacube is associated with data collected to support another operation. It is also data whose confidentiality, integrity, and availability need to be assured. The lack of these information assurance attributes is cause for an alert to the operators, as it may compromise a mission that the datacube supports. Information describing the situational awareness of the datacube information assurance is visualized at this system. For example, during a HA/DR operation in Singapore, social media data may need to be collected from the public in order to assess the state of civilians in that region, since it is documented that citizens use social media resources to communicate during disasters (IDA 2013). In collecting and analyzing such data, it is important to visualize the confidentiality, integrity, and availability of the datacubes in order to be assured that the troops assisting in the operation are provided accurate, concise, and relevant information.

b. Cyber Attack Detection Situational Awareness (Cyber Attack Detection SA) System

The Cyber Attack Detection Situational Awareness (CADSA) System obtains the configuration parameters from the overarching Cyber SA System and continuously monitors for signs of a cyber attack; it will alert the main cloud automatically in the event of an attack. In addition, alerts will be sent to the IASA cloud service in the form of additional monitoring parameters to heighten the information assurance level. It is important to be alerted on attempted cyber attacks on the infrastructure, including the datacubes being used. It is important to have situational awareness of the datacube's vulnerability of being exploited. For example, in a rescue operation, if a hacker attacks the rescue support systems supporting rescue operations, in addition to destroying infrastructure, he or she may steal or modify the social media data that was collected in order to mislead the rescue personnel.

c. Information Assurance Situational Awareness (Information Assurance SA) System

Finally, the IASA System obtains the configuration parameters from the overarching Cyber SA system and continuously monitors specified files (datacubes) on specified devices for changes made. If the datacube performs a behavior that was of interest to the operator and is indicative of loss of confidentiality, integrity, or availability, all operators who subscribed to that behavior (via the Cyber SA System) are alerted automatically once trigger events occur. In addition, alerts will also be sent to the CADSA cloud system to highlight detected information assurance behaviors compared to the standard behaviors. This action heightens the alert level for monitoring of cyber attacks within the network infrastructure. This interdependent relationship among the three cloud systems allows higher synergy and information sharing.

B. SYSTEM OPERATIONAL USE CASES

1. Scenario 1: Operator Input Configuration Setting to System

The operator enters the inputs desired for the network/files to be monitored by Cyber Situational Awareness and Command and Control (CSAC2). CSAC2 receives inputs from the operator and determines if the input is classified as Case 1 and/or Case 2. Under Case 1, CSAC2 generates configuration files to Information Assurance Situational Awareness (IASA). IASA receives configuration files and checks configuration files for a security key or encrypted operator signature. Once the configuration files are accepted, IASA will adjust the monitoring areas. Under Case 2, CSAC2 generates configuration files to Cyber Attack Detection Situational Assurance (CADSA). CADSA receives configuration files and checks configuration files for a security key or encrypted operator signature. Once the configuration files are accepted, CADSA will adjust monitoring network.

2. Scenario 2: CADSA Detects an Abnormality in the Network

CADSA detects an abnormality in the monitored port(s) and sends an alert to CSAC2. CSAC2 receives the alert and checks for a security key or encrypted signature. Once the alert is found to be authentic, CSAC2 alerts the operator (see scenario 6). CADSA also sends an alert to IASA to increase the frequency of monitoring of files or to increase the number of files to monitor. IASA receives the alert, checks for a security key or encrypted signature, and increases the number of files to be monitored or the frequency of monitoring of files. CADSA will self-configure to increase its monitoring rate of the network.

3. Scenario 3: CADSA Detects an External Attack

CADSA detects an attack at port number XX and sends an alert to CSAC2. CSAC2 receives the alert and checks for a security key or encrypted signature. Once the alert is found to be authentic, CSAC2 alerts the operator (see scenario 6). CADSA sends an alert to IASA to increase the frequency of monitoring of files or to increase the number of files to monitor. IASA receives the alert, checks for a security key or encrypted signature, and encrypts high-priority files which are predetermined by the operator. CADSA will close the attacked port(s) and increase the monitoring rate of the network to achieve heightened security level status.

4. Scenario 4: IASA Detects an Abnormality in the Network

IASA detects an abnormality in one of the monitored files or directories, and IASA sends an alert to CSAC2. CSAC2 receives the alert and checks for a security key or encrypted signature. Once the alert is found to be authentic, CSAC2 alerts the operator (see scenario 6). IASA sends an alert to CADSA to increase the monitoring rate of the network. CADSA receives the alert, checks for a security key or an encrypted signature and increases the monitoring rate of the network. IASA will self-configure to increase the frequency of file monitoring or the number of files to be monitored.

5. Scenario 5: IASA Detects Internal Unauthorized Changes to High Priority Files

IASA detects unauthorized changes to high priority files or unauthorized access to high security level folders and sends an alert to CSAC2. CSAC2 receives the alert and checks for a security key or encrypted signature. Once the alert is found to be authentic, CSAC2 alerts the operator (see scenario 6). IASA sends an alert to CADSA to increase the monitoring rate of the network to achieve heightened security level status. CADSA receives the alert, checks for a security key or an encrypted signature, and increases the monitoring rate of the network. IASA will encrypt high priority files which are predetermined by operator.

6. Scenario 6: CSAC2 Visualizes Alerts to the Operator

CSAC2 receives alerts from CADSA/IASA and displays alerts to the operator.

C. INTELLIGENCE AUTOMATION APPLIED TO INFORMATION ASSURANCE

This section reviews the technologies or methodologies used to automate information assurance.

1. Monitored Targets To Automate Behavior Analysis

Recall that data can be in the form three states, namely, data at rest, data in transit and data in use. Suitable sensors must be deployed and used in order to correctly monitor and capture the status of the data in terms of their confidentiality, integrity and availability.

2. Cyber Sensors To Sense for Targets

There are two ways of deploying cyber sensors in general. They are classified by where are the monitoring is required—that is, monitoring of the network or monitoring of network devices, also known as the host. Therefore, cyber sensors can be generally categorized by the location in which they have been deployed, either network based or host based (Deptula 2013).

Specific to automating IA, network-based cyber sensors are used to monitor information in the state of data in transit, while host-based cyber sensors are used to monitor information in the state of data at rest and data in use.

Furthermore, automation of information assurance requires the data to be monitored for loss of confidentiality, integrity, and availability. Additionally, since data changes states between data at rest, data in transit, and data in use, there is a need to monitor the data's loss of confidentiality, integrity, and availability for each state. Table 1 provides the nine different combinations of IA across the three different states of data (S_{ij}), where i represents the IA condition and j represents the state of data.

		Information Assurance Condition		
		Confidentiality	Integrity	Availability
Data States	At Rest	S_{11}	S_{21}	S_{31}
	In Transit	S_{12}	S_{22}	S_{32}
	In Use	S_{13}	S_{23}	S_{33}

Table 1. Nine combinations of information assurance at different states of data.

This thesis will specifically focus on hosts with the Microsoft (MS) Windows operating system for the data at rest state, namely S_{11}, S_{21}, and S_{31}. The remaining sections discuss background on cyber sensors used to assure the confidentiality, integrity, and availability of data in MS Windows for the data at rest state.

a. Sensors for MS Windows

There are numerous readily available methods or applications that can be used as sensors to automate information assurance of data in its three states; they can be grouped into the type of operation systems or platforms and the data states. The main focus will be on sensors that are used in Microsoft's operating system and data in storage and the in-use state. Sensors to monitor data in transit will not be covered. Commands found in

Microsoft's DOS and MATLAB program can be used to monitor data in the storage state, and the program Process Explorer by Mark Russinovich can be used for data in the in-use state.

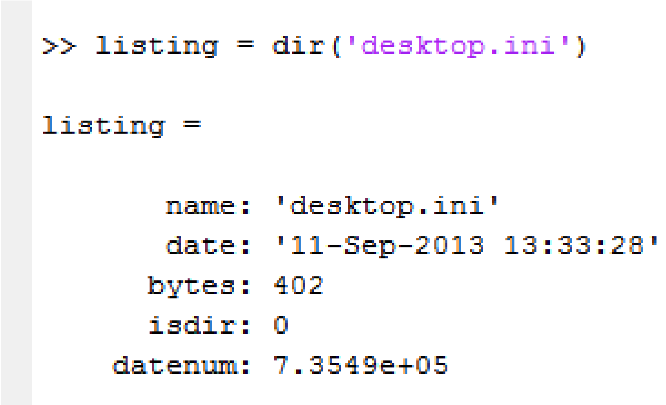
(1) MS DOS Command–DIR. The DIR command in Microsoft's DOS provides a list of the available contents of a directory with information including the data's last modification date and time, the file size, and owner. It offers the option to search through a specified directory to display files of interest.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N] [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

/Q – Display the owner of the file.

/S - Displays files in specified directory and all subdirectories (Search)

(2) MATLAB Command–DIR. The DIR command in MATLAB has functions similar to the DIR command in DOS. It provides the file and folder names in the MATLAB current folder, last modification date and time, and file size. It is commonly used with the command LISTING to obtain attributes of the file, an example of the result of the command is shown in Figure 25.

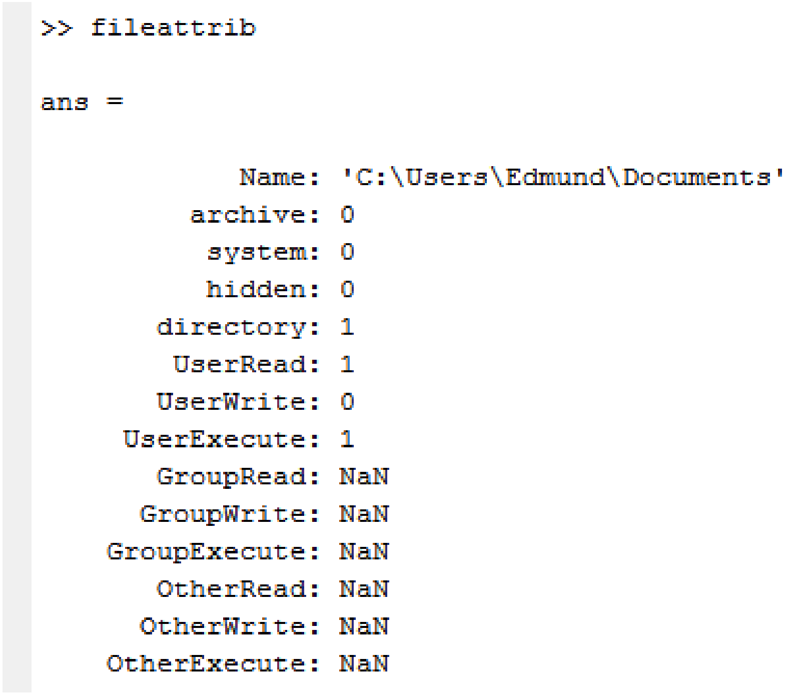


```
>> listing = dir('desktop.ini')

listing = 

       name: 'desktop.ini'
       date: '11-Sep-2013 13:33:28'
      bytes: 402
      isdir: 0
    datenum: 7.3549e+05
```

Figure 25. Typical result of the DIR command in MATLAB.

(3) MATLAB Command–FILEATTRIB. FILEATTRIB is another command in MATLAB that provides information regarding a file. FILEATTRIB gets attribute values for the current folder, using the structure shown in Figure 26, where Name is

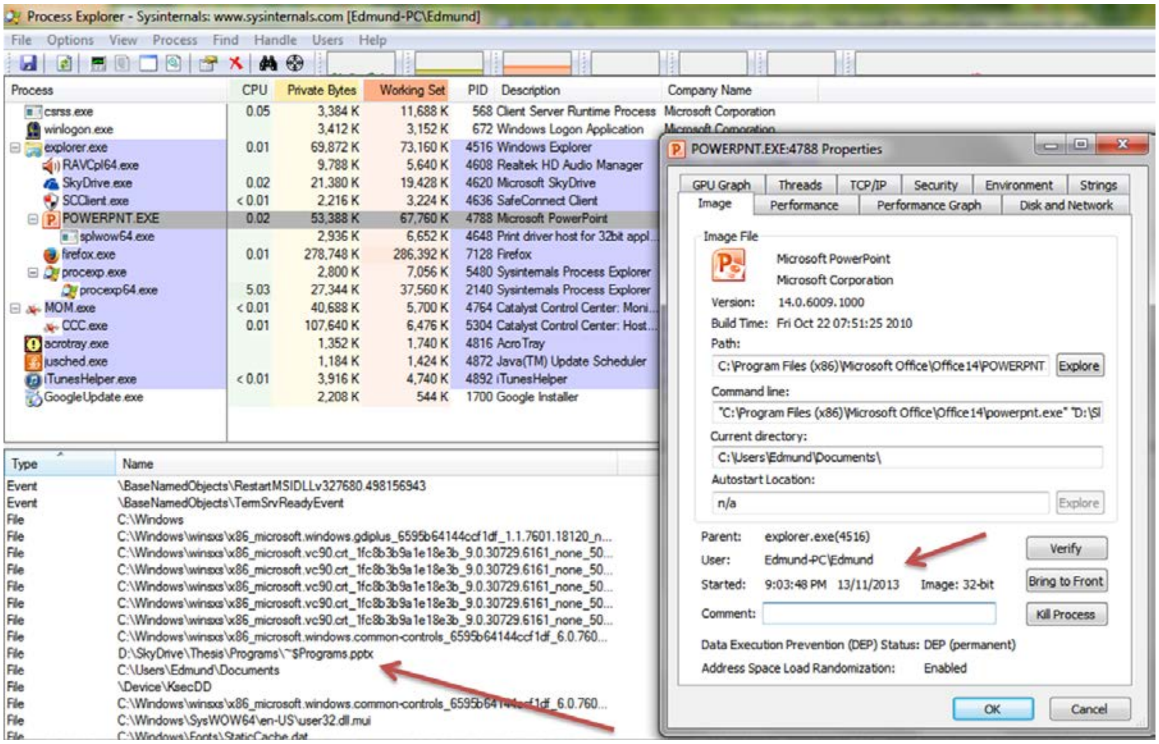
always a string containing the current folder name. For the other fields, a value of 0 indicates that the attribute is off, 1 indicates that the attribute is on, and NaN indicates that the attribute does not apply.



```
>> fileattrib

ans =

             Name: 'C:\Users\Edmund\Documents'
          archive: 0
           system: 0
           hidden: 0
        directory: 1
         UserRead: 1
        UserWrite: 0
      UserExecute: 1
        GroupRead: NaN
       GroupWrite: NaN
     GroupExecute: NaN
        OtherRead: NaN
       OtherWrite: NaN
     OtherExecute: NaN
```

Figure 26. Typical result of the FILEATTRIB command in MATLAB.

(4) Process Explorer. Process Explorer is a program used to monitor data that are in use. It lists information regarding the user, the time the data is accessed, and the files used by the program (Figure 27). However, information regarding the location where the data is temporarily stored in memory is not available.



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
csrss.exe	0.05	3,384 K	11,688 K	568	Client Server Runtime Process	Microsoft Corporation
winlogon.exe		3,412 K	3,152 K	672	Windows Logon Application	Microsoft Corporation
explorer.exe	0.01	69,872 K	73,160 K	4516	Windows Explorer	
RAVCpl64.exe		9,788 K	5,640 K	4608	Realtek HD Audio Manager	
SkyDrive.exe	0.02	21,380 K	19,428 K	4620	Microsoft SkyDrive	
SCClient.exe	< 0.01	2,216 K	3,224 K	4636	SafeConnect Client	
POWERPNT.EXE	0.02	53,388 K	67,760 K	4788	Microsoft PowerPoint	
splwow64.exe		2,936 K	6,652 K	4648	Print driver host for 32bit appl...	
firefox.exe	0.01	278,748 K	286,392 K	7128	Firefox	
procexp.exe		2,800 K	7,056 K	5480	Sysinternals Process Explorer	
procexp64.exe	5.03	27,344 K	37,560 K	2140	Sysinternals Process Explorer	
MOM.exe	< 0.01	40,688 K	5,700 K	4764	Catalyst Control Center: Moni...	
CCC.exe	0.01	107,640 K	6,476 K	5304	Catalyst Control Center: Host...	
acrotray.exe		1,352 K	1,740 K	4816	AcroTray	
jusched.exe		1,184 K	1,424 K	4872	Java(TM) Update Scheduler	
iTunesHelper.exe	< 0.01	3,916 K	4,740 K	4892	iTunesHelper	
GoogleUpdate.exe		2,208 K	544 K	1700	Google Installer	

Type	Name
Event	\BaseNamedObjects\RestartMSIDLLv327680.498156943
Event	\BaseNamedObjects\TermSrvReadyEvent
File	C:\Windows
File	C:\Windows\winsxs\x86_microsoft.windows.gdiplus_6595b64144ccf1df_1.1.7601.18120_n...
File	C:\Windows\winsxs\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50...
File	C:\Windows\winsxs\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50...
File	C:\Windows\winsxs\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50...
File	C:\Windows\winsxs\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50...
File	C:\Windows\winsxs\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50...
File	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.760...
File	D:\SkyDrive\Thesis\Programs\~$Programs.pptx
File	C:\Users\Edmund\Documents
File	\Device\KsecDD
File	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.760...
File	C:\Windows\SysWOW64\en-US\user32.dll.mui
File	C:\Windows\Fonts\StaticCache.dat

Image File

Microsoft PowerPoint

Microsoft Corporation

Version: 14.0.6009.1000

Build Time: Fri Oct 22 07:51:25 2010

Path: C:\Program Files (x86)\Microsoft Office\Office14\POWERPNT

Command line: "C:\Program Files (x86)\Microsoft Office\Office14\powerpnt.exe" "D:\Sl

Current directory: C:\Users\Edmund\Documents\

Autostart Location: n/a

Parent: explorer.exe(4516)

User: Edmund-PC\Edmund

Started: 9:03:48 PM 13/11/2013 Image: 32-bit

Comment:

Data Execution Prevention (DEP) Status: DEP (permanent)

Address Space Load Randomization: Enabled

Figure 27. Output display of Process Explorer.

3. Detect, Identify, Predict, and React (DIPR) Intelligence Automation for Data Behavior Analysis

The concept of using the DIPR model to achieve information assurance of data is feasible. Information and status regarding the targeted data can be extracted and monitored by sensor software that resides or is deployed in the storage hosts or networks. Different types of sensors are required depending on the targeted data state (at rest, in transit, or in use) and the operating system in which the data resides. The focus of this thesis will be mainly on monitoring data in the at-rest state (storage) and operating in an MS Windows environment, using tools designed for operating in the MS Windows system. How the DIPR model is used to process cyber sensor data in order to look for unauthorized behavior of the data is discussed in the functional decomposition of the Information Assurance Situational Awareness system within this chapter.

D. EXTERNAL SYSTEMS DIAGRAM

Figure 28 provides the External Systems Diagram (content diagram) for Cyber Situational Awareness & Information Assurance of Smart Cloud System of Systems (The System). The System (A0) interacts with four external systems: the operators (B1), Smart Cloud SoS (B2), Government Cyber Policy Department (B3), and the External Infrastructures (B4).

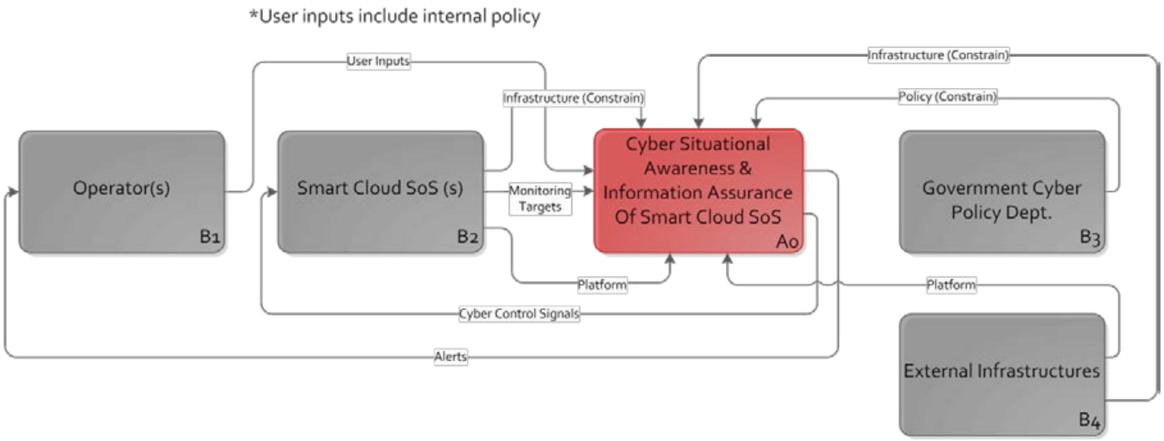

*User inputs include internal policy
User Inputs
Infrastructure (Constrain)
Infrastructure (Constrain)
Policy (Constrain)
Operator(s)
B1
Smart Cloud SoS (s)
B2
Monitoring Targets
Cyber Situational Awareness & Information Assurance Of Smart Cloud SoS
A0
Government Cyber Policy Dept.
B3
Platform
Platform
Cyber Control Signals
Alerts
External Infrastructures
B4

Figure 28. External system diagram for the system.

The operations are the personnel operating the system and overseeing the system's command and control function. The smart cloud SoS is the external info-communication system in which the operator has a stake and interest; it is also the system whereby the sensors from The System will be deployed to monitor the information assurance of the data and network security. The Government Cyber Policy Department is the government body which has a stake and interest in the smart cloud SoS and provides the overall governing policies in the cyber environment. External Infrastructures provide the platforms, environments, hardware, and network architecture in which the entire system of systems resides and operates.

There are two types of external input to The System. They are monitoring targets from the smart cloud SoS(s) and user inputs from the operator(s). Monitoring targets are the information data and network from the smart cloud SoS in which the operator is interested in ensuring situation awareness and information assurance. User inputs are the command and control/operational instructions from the operator to The System to determine how The System operates in the overall cyber environment. It also consists of the operator's internal policies on how The System interacts with internal components and externally.

There are two types of external output from The System. They are the cyber control signals to the smart cloud SoS(s) and alerts to the operator(s). Cyber control signals are the control instructions to the smart clouds where The System's sensors are deployed; it provides the signal to encrypt files/folders of interest or to heighten the network security level due to the detection of an abnormality in the environment in which the information confidentiality, integrity, and availability are degraded. Alerts are output from The System to the operator when degradation of information confidentiality, integrity, and availability is detected in the environment.

There are three types of constraints on The System. They include constraints from the infrastructure of the smart cloud SoS and external infrastructures and policies from the Government Cyber Control Department. The operating environment (software/OS) in which the smart cloud SoS operates will constrain the type of sensors that can be deployed in the cloud. The external infrastructures of the entire cyber environment will determine the limitations and constraints that The System will face, such as the hardware and networking architecture used. The cyber policies set by the government will determine the boundaries or limitations of how The System operates and interacts with external bodies.

There are two types of enablers: the platform where the sensors will be deployed in the smart clouds and the platform of the external infrastructures which determine how The System will operate and function in the cyber environment.

E. FUNCTIONS DEFINITION AND DECOMPOSITION

The Functional Hierarchy of the system, as shown in Figure 29, "Provide Cyber Situational Awareness & Information Assurance of Smart Cloud SoS" consists of four main functions, namely "Provide Command & Control (A1)," "Provide Situational Awareness (A2)," "Provide Information Assurance SA (A3)," and "Provide Cyber Attack

Detection SA (A4).” Function A1 and A2 will form the main interface whereby the command and control (C2) and the feedback from the alerts will be provided to the operators on the status of the systems.

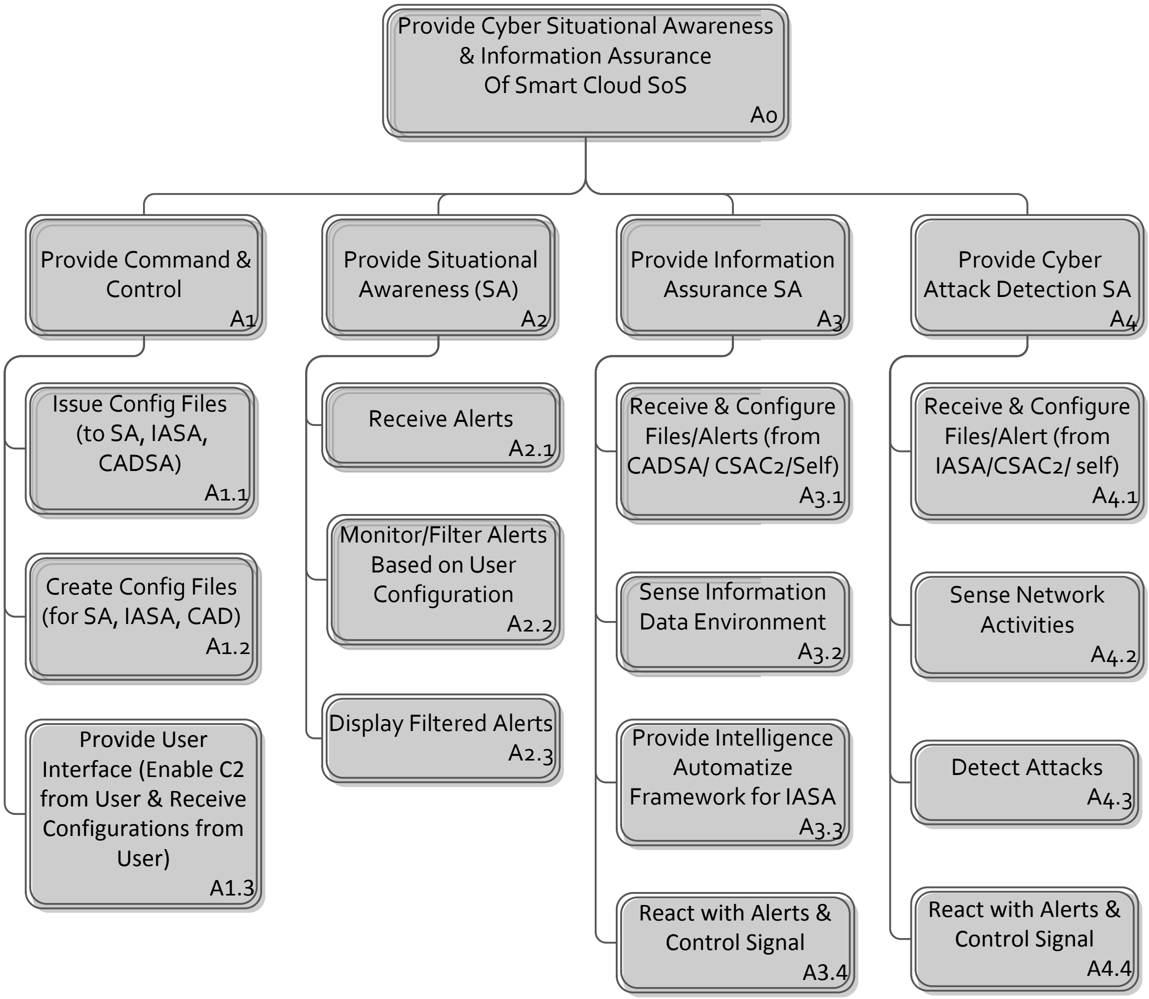

Provide Cyber Situational Awareness & Information Assurance Of Smart Cloud SoS A0
Provide Command & Control A1
Provide Situational Awareness (SA) A2
Provide Information Assurance SA A3
Provide Cyber Attack Detection SA A4
Issue Config Files (to SA, IASA, CADSA) A1.1
Create Config Files (for SA, IASA, CAD) A1.2
Provide User Interface (Enable C2 from User & Receive Configurations from User) A1.3
Receive Alerts A2.1
Monitor/Filter Alerts Based on User Configuration A2.2
Display Filtered Alerts A2.3
Receive & Configure Files/Alerts (from CADSA/ CSAC2/Self) A3.1
Sense Information Data Environment A3.2
Provide Intelligence Automatize Framework for IASA A3.3
React with Alerts & Control Signal A3.4
Receive & Configure Files/Alert (from IASA/CSAC2/ self) A4.1
Sense Network Activities A4.2
Detect Attacks A4.3
React with Alerts & Control Signal A4.4

Figure 29. Functional hierarchy of the system.

Within the function A1, three sub-functions allow the operators to create and issue configuration/parameter files to the other subsystem (SoS) and the interface to interact with the operators. Together these functions enable the command and control of the SoS. The three sub-functions of A1 are “Issue Config Files (A1.1),” “Create Config File (A1.2),” and “Provide User Interface (A1.3).”

Within the function A2, three sub-functions provide the operators with the current status or situation awareness of the system/network based on the configurations set by the operators, by receiving filtered alerts from the two subsystems, A3 & A4. The three sub-functions of A2 are “Received Alerts (A2.1),” “Monitor/Filter Alerts Based on User Configuration (A2.2),” and “Display Filtered Alerts (A2.3).”

Within the function A3, four sub-functions monitor the files/folders of interest selected by the operators for suspicious activities and unauthorized changes to the files/folders. The sub-functions also issue alerts to subsystems A2 & A4 and the control signal to encrypt the files/folders. The four sub-functions of A3 are “Received & Configure Files/Alerts (A3.1),” “Sense Information Data Environment (A3.2),” “Provide Intelligence Automatize Framework for IASA (A3.3),” and “React with Alerts & Control Signal (A3.4)”.

Within the function A4, four sub-functions monitor the networks based the configurations set by the operators for suspicious activities and attacks on the network. The sub-functions issue alerts to subsystems A2 & A3 and the control signal to counter/control the attacks. The four sub-functions of A4 are “Received & Configure Files/Alerts (A4.1),” “Sense Network Activities (A4.2),” “Detect Attacks (A4.3),” and “React with Alerts & Control Signal (A4.4).”

The function “Sense Information Data Environment (A3.2)” consists of three sub-functions to sense the data based on its state (Figure 30). The three states are data in stored (data at rest) condition on storage devices, such as a hard disk; data in transit state through transmission on the network system, such as transferring the data on WIFI or LAN; and the final state when the data is in use by the user where the data is residing in memory, such as RAM on the user’s device.

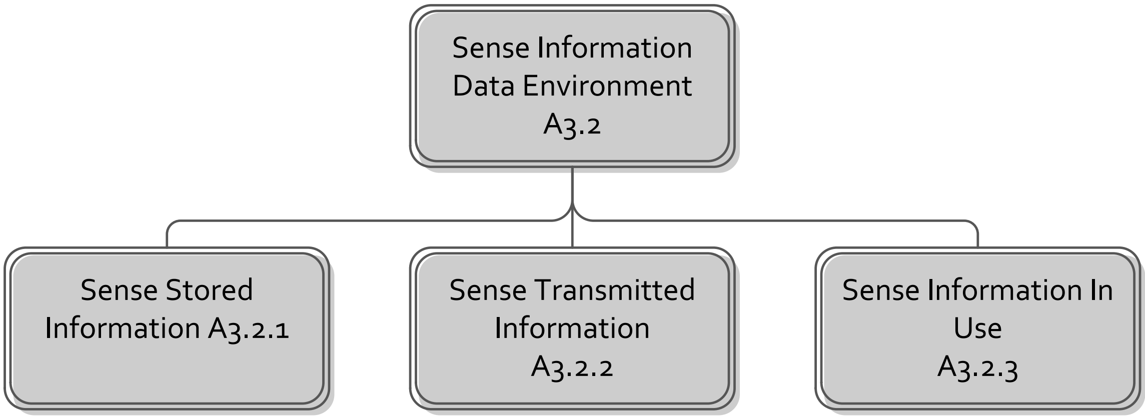

Sense Information Data Environment A3.2
Sense Stored Information A3.2.1
Sense Transmitted Information A3.2.2
Sense Information In Use A3.2.3

Figure 30. Functional hierarchy of the function "Sense Information Data Environment."

The function "Provide Intelligence Automatize Framework for IASA (A3.3)" consists of four sub-functions that make up the DIPR process (Figure 31). DIPR is the key component/feature to provide smart automation by filtering the large amount of data collected from the sensors to manageable information that are critical to the operator, or reaction as countermeasure. This study will be focusing on the Detect and Identify framework of this function.

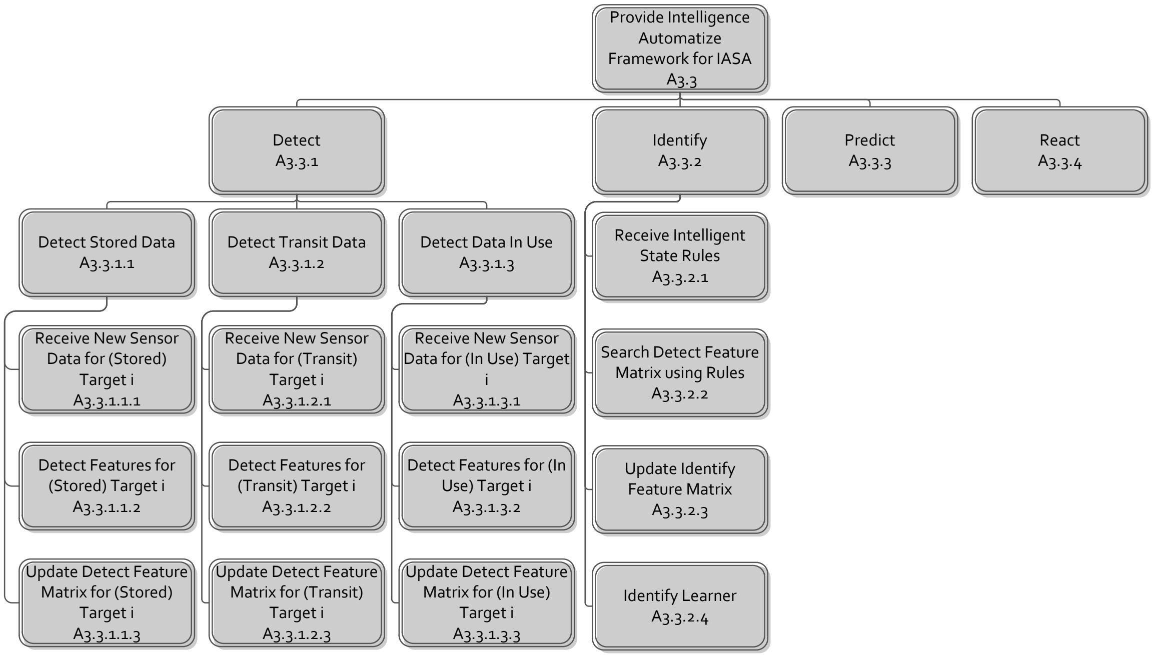

Provide Intelligence Automatize Framework for IASA A3.3
Detect A3.3.1
Identify A3.3.2
Predict A3.3.3
React A3.3.4
Detect Stored Data A3.3.1.1
Detect Transit Data A3.3.1.2
Detect Data In Use A3.3.1.3
Receive Intelligent State Rules A3.3.2.1
Receive New Sensor Data for (Stored) Target i A3.3.1.1.1
Receive New Sensor Data for (Transit) Target i A3.3.1.2.1
Receive New Sensor Data for (In Use) Target i A3.3.1.3.1
Search Detect Feature Matrix using Rules A3.3.2.2
Detect Features for (Stored) Target i A3.3.1.1.2
Detect Features for (Transit) Target i A3.3.1.2.2
Detect Features for (In Use) Target i A3.3.1.3.2
Update Identify Feature Matrix A3.3.2.3
Update Detect Feature Matrix for (Stored) Target i A3.3.1.1.3
Update Detect Feature Matrix for (Transit) Target i A3.3.1.2.3
Update Detect Feature Matrix for (In Use) Target i A3.3.1.3.3
Identify Learner A3.3.2.4

Figure 31. Functional hierarchy of the function "Provide Intelligence Automatize Framework for IASA."

Depending on the category of the monitored data, the sub-function of “Detect (A3.3.1)” receives data from sensors deployed on external smart clouds, detects and extracts relevant information from the data, and updates the target detect feature matrix. “Identify (A3.3.2)” looks for changes in the feature matrix based on intelligent state rules using its four sub-functions, “Receive Intelligent State Rule (A3.3.2.1),” “Search Detect Feature Matrix using Rules (A3.3.2.2),” and “Update Identify Feature Matrix (A3.3.2.3).” The function “Identify Learner (A3.3.2.4)” operates asynchronously from the main identify functions to learn and set/change intelligent state rules.

F. IDEF0 DIAGRAM OF PROPOSED SYSTEM

1. Cyber Situational Awareness and Information Assurance of Smart Cloud SoS IDEF0 (A0 Level 1)

This level 1 IDEF0 diagram (Figure 32) depicts the level 1 functions of The System (A0). As detailed previously in the External Systems Diagram description, there are several external inputs, outputs, constraints, and enablers that interact with The System.

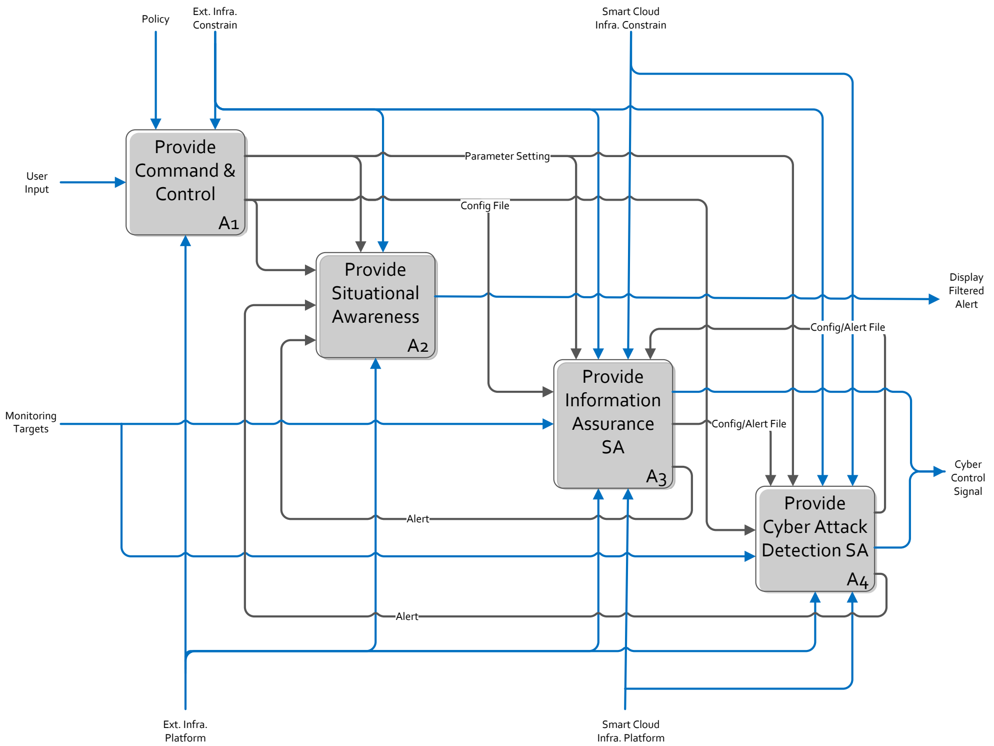


Figure 32. Level 1 IDEF0 diagram of the system.

Figure 32 specifies which level 1 functions of The System the external inputs, outputs, constraints, and enablers interact with. Interaction is defined as the transfer of EMMI (Energy, Material, Material Wealth, Information) to the external systems (indicated in blue). Monitoring targets from the smart cloud SoS provides inputs to both the “Provide Information Assurance SA (A3)” and “Provide Cyber Attack Detection SA (A4)” functions. Both functions A3 and A4 will receive the targets of interest for the deployment of smart sensors. User Inputs from the Operators go to “Provide Command & Control (A1),” which will receive and configure the system based on the inputs.

The Policy constraint sets how The System functions and is received by “Provide Command & Control (A1),” which creates the configuration files and parameter settings based on the policy given. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also among each other within The System. The External Infrastructures constraint goes to all four functions (A1 to A4). The Smart Cloud Infrastructure constraint only affects functions “Provide Information Assurance SA (A3)” and “Provide Cyber Attack Detection SA (A4)” as the type of operating system used in different smart clouds will determine the type of sensors being deployed.

Output in the form of filtered alerts will be given by the “Provide Situation Awareness (A2)” function. All alerts from A3 and A4 will be intelligently analyzed, filtered, and summarized before creating alerts as output to the Operators (B1). Output as reaction to the current situation (threat or attack) will be given by the “Provide Information Assurance SA (A3)” and “Provide Cyber Attack Detection SA (A4)” functions in the form of cyber control signals/instructions to smart clouds to counter or react to the situation.

The two types of enablers, the platform where the sensors will be deployed in the smart clouds (for A3 & A4 only) and the platform of the external infrastructures, determine how all the functions will operate in the cyber environment. Figure 32 also specifies how the level 1 functions of The System interact with each other. Interaction or the transfer of EMMI within the system/level is indicated in black. The Parameter Setting from “Provide Command & Control (A1),” which is determined by the policy constraint, will indirectly impact and determine the operational constrains for functions A2 to A4. Based on the policy constraint given, configuration files created by “Provide Command & Control (A1)” will be the input for the functions A2 to A4. With the parameter setting and configuration files from A1, “Provide Information Assurance SA (A3)” and “Provide Cyber Attack Detection SA (A4)” functions will each create alerts based on the situation encountered. The Alert files from A3 & A4 will be input for A2. The A3 and A4 functions also support each other with configuration files to ‘alert’ them to the current situation and advise on the need to increase the monitoring level.

2. Information Assurance Situational Awareness IDEF0 (A3 Level 2)

This level 2 IDEF0 diagram (Figure 33) depicts the functions of “Provide Information Assurance SA (A3).” As detailed previously with the IDEF0 diagram of The System (A0) description, there are several external inputs, outputs, constraints, and enablers that interact with The System.

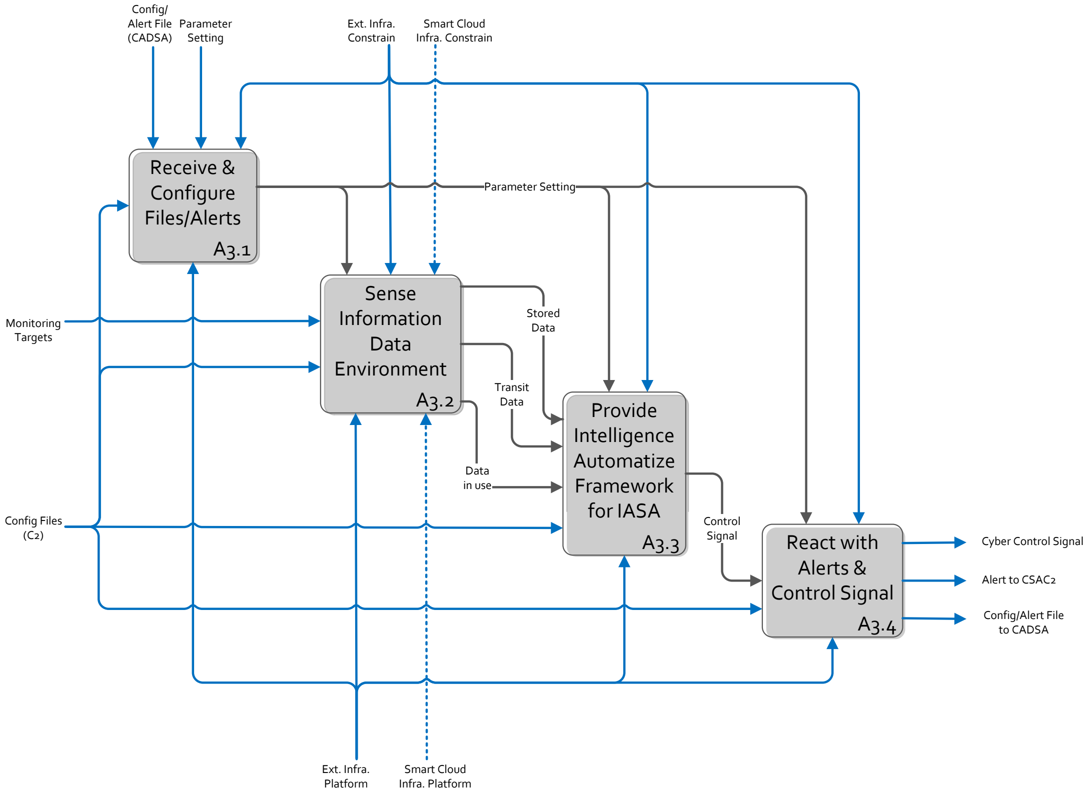

Config/ Alert File (CADSA)
Parameter Setting
Ext. Infra. Constrain
Smart Cloud Infra. Constrain
Receive & Configure Files/Alerts
A3.1
Parameter Setting
Sense Information Data Environment
A3.2
Stored Data
Transit Data
Data in use
Provide Intelligence Automatize Framework for IASA
A3.3
Control Signal
React with Alerts & Control Signal
A3.4
Monitoring Targets
Config Files (C2)
Cyber Control Signal
Alert to CSAC2
Config/Alert File to CADSA
Ext. Infra. Platform
Smart Cloud Infra. Platform

Figure 33. Level 2 IDEF0 diagram of the function “Provide Information Assurance SA.”

Figure 33 specifies which level 2 functions of A3 the external inputs, outputs, constraints, and enablers interact with. Interaction is defined as the transfer of EMMI to the external systems (indicated in blue). Data from monitored targets from the smart cloud SoS inputs to the "Sense Information Data Environment (A3.2)" function for analysis and filtering to the three different states in which the data can reside. Configuration files created by the A1 function will serve as inputs to the sub-functions of A3 and determine their operation environment.

Configuration files created by the A4 function will provide additional configuration/constraints as a form of reaction due to the detected situation. The Parameter Setting constraint sets the A3 function and is received by "Receive & Configure Files/Alerts (A3.1)," which creates the parameter settings for the other sub-functions of A3 (A3.2 to A3.4). The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all four functions (A3.1 to A3.4). The Smart Cloud Infrastructure constraint only affects the function "Sense Information Data Environment (A3.2)" as the type of operating system used in different smart clouds will determine the type of data collected for analysis.

From the situation gathered and analyzed by the A3.3 function in the form of a control signal, function A3.4 will create the required output to external functions in the form of cyber control signals for other functions in the system, alerts to CSAC2 for alerts to operators, and configuration/alert files to CADSA to increase synergy and information sharing between systems.

The two types of enablers, the platform where the sensors will be deployed in the smart clouds (for A3.2 only) and the platform of the external infrastructures, determine how all the functions will operate in the cyber environment.

Figure 33 also specifies how the level 2 functions of A3 interact with each other. Interaction or the transfer of EMMI within the system/level is indicated in black. The Parameter Setting from "Receive & Configure Files/Alerts (A3.1)," determined by the parameter setting and configuration/alert files from CADSA, will affect and determine the operational environment of functions A3.1 to A3.3. Collected data from the targets will be broken down by "Sense Information Data Environment (A3.2)" function as three types of inputs for "Provide Intelligence Automotive Framework for IASA (A3.3)." The three types are Stored Data (Data at Rest), Transit Data (Data in Motion), and Data In Use, which encompass the entire range/state of the data/information. Data processed or analyzed by the A3.3 function will generate the required control signal to the "React with Alerts & Control Signal (A3.4)" for reaction to the situation.

3. Sense Information Data Environment IDEF0 (A3.2 Level 3)

This level 3 IDEF0 diagram (Figure 34) depicts the functions of the "Sense Information Data Environment (A3.2)." As detailed previously with the IDEF0 diagram of the function "Provide Information Assurance SA (A3)" description, there are several external inputs, outputs, constraints, and enablers that interact with A3.2.

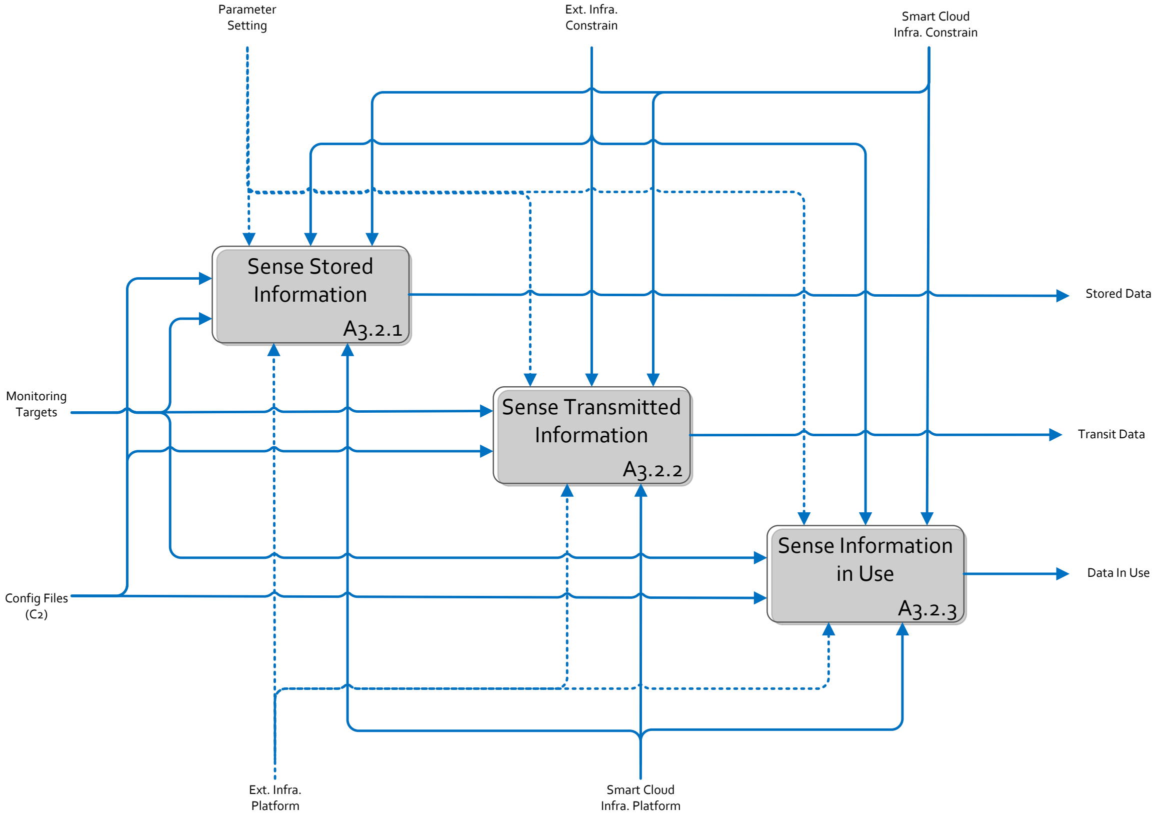


Figure 34. Level 3 IDEF0 diagram of function "Sense Information Data Environment."

This diagram specifies with which level 3 functions of A3.2 the external inputs, outputs, constraints, and enablers interact. Interaction is defined as the transfer of EMMI to the external systems, as indicated in blue.

Data from monitored targets from the smart cloud SoS inputs to the "Sense Stored Information (A3.2.1)," "Sense Transit Information (A3.2.2)," and "Sense Information In Use (A3.2.3)" functions for analysis. Configuration files created by the A1 function will serve as inputs to sub-functions of A3.2 and determine their operation environment.

The Parameter Setting constraint from the A3.1 function is received by all sub-functions (A3.2.1 to A3.2.3) of A3.2 and will set how the sub-functions operate. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all four functions (A3.2.1 to A3.2.3). The Smart Cloud Infrastructure constraints will affect all sub-functions (A3.2.1 to A3.2.3) as the type of operating system used in different smart clouds will determine the type of data collected for analysis.

The data gathered from the targets is sorted according to the type of sensors deployed in the smart clouds by the sub-functions A3.2.1 to A3.2.3. This data will be identified as three types of output for "Provide Intelligence Automotive Framework for IASA (A3.3)"—Stored Data (Data at Rest), Transit Data (Data in Motion), and Data In Use, which encompass the entire range/state of data/information.

The two types of enablers, the platform where the sensors will be deployed in the smart clouds and the platform of the external infrastructures, determine how all the functions will operate in the cyber environment.

There is no internal interaction within the A3.2 function.

4. Sense Stored/Transit/In Use Information IDEF0 (A3.2.X Level 4)

This level 4 IDEF0 diagram (Figure 35) depicts the functions of the "Sense Stored/Transit/In Use Information (A3.2.X)." The X represents the function numbering based on the type of data being processed, sensing of stored data, and whether the data is in transit or in use. These are numbered 1, 2, and 3, respectively. As detailed previously with the IDEF0 diagram of the function “Sense Information Data Environment (A3.2)” description, there are several external inputs, outputs, constraints, and enablers that interact with A3.2.X.

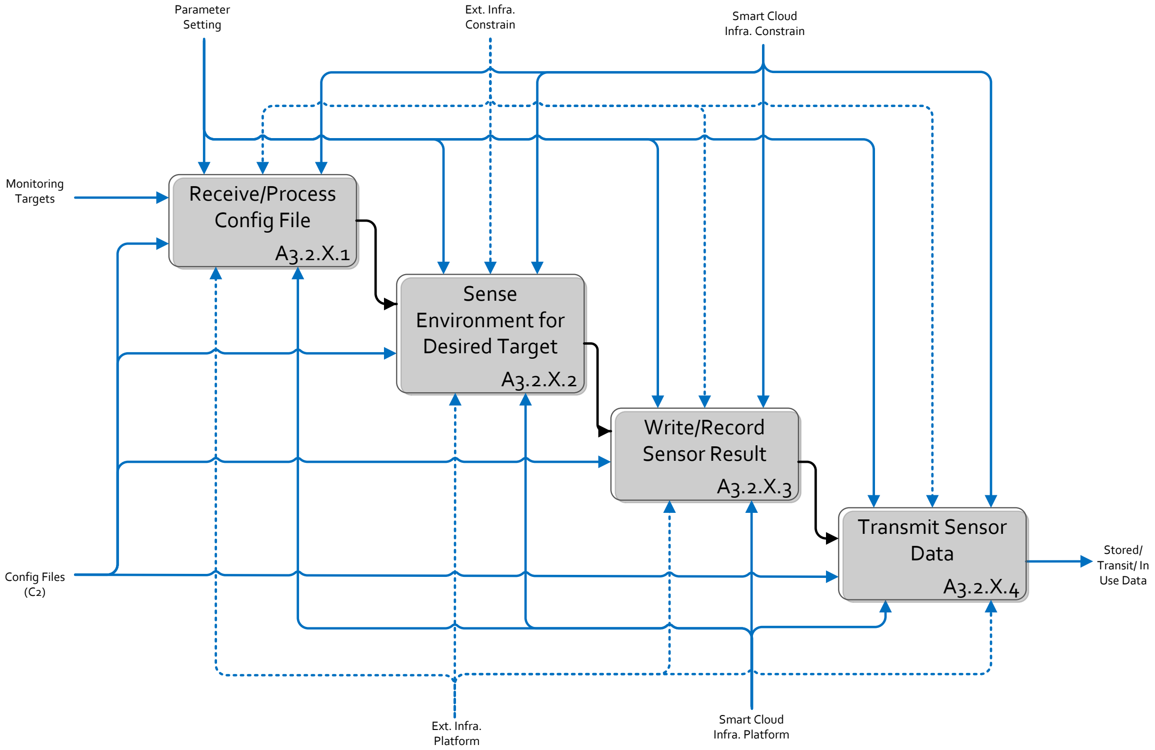

Parameter Setting
Ext. Infra. Constrain
Smart Cloud Infra. Constrain
Monitoring Targets
Receive/Process Config File
A3.2.X.1
Sense Environment for Desired Target
A3.2.X.2
Write/Record Sensor Result
A3.2.X.3
Transmit Sensor Data
A3.2.X.4
Stored/ Transit/ In Use Data
Config Files (C2)
Ext. Infra. Platform
Smart Cloud Infra. Platform

Figure 35. Level 4 IDEF0 diagram of function “Sense Stored/Transit/In Use Information.”

This diagram specifies with which level 4 functions of A3.2.X the external inputs, outputs, constraints, and enablers interact. Interaction is defined as the transfer of EMMI to the external systems, as indicated in blue. The function “Receive/Process Config File (A3.2.X.1)” receives and processes the sensor data collected from monitored targets from the smart cloud SoS. Configuration files created by the A1 function will serve as inputs to sub-functions of A3.2.X and determine their operation environment.

The Parameter Setting constraint from the A3.1 function is received by all sub-functions (A3.2.X.1 to A3.2.X.4) of A3.2.X and will set how the sub-functions operate. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all four functions (A3.2.X.1 to A3.2.X.4). The Smart Cloud Infrastructure constraint affects all functions of A3.2.X as the data collected in different smart clouds will determine the type of data analyzed.

From the sensor data gathered, after filtering, processing, and recording by the functions of A3.2.X, function A3.2.X.4 will create the required output to A3.3 for DIPR analysis.

The two types of enablers, the platform where the sensors will be deployed in the smart clouds and the platform of the external infrastructures, determine how all the functions will operate in the cyber environment.

This diagram also specifies how the level 4 functions of A3.2.X interact with each other. Interaction or the transfer of EMMI within the system/level is indicated in black. Data on the target are first received by "Receive/Process Config File (A3.2.X.1)." The data are then analyzed, filtered, and sorted by "Sense Environment for Desired Target (A3.2.X.2)" before the result is recorded by the function "Write/Record Sensor Result (A3.2.X.3)." The results are transmitted to A3.3 for DIPR analysis by the function "Transmit Sensor Data (A3.2.X.4).

5. Intelligence Automatize Framework for IASA IDEF0 (A3.3 Level 3)

This level 3 IDEF0 diagram (Figure 36) depicts the functions of "Provide Intelligence Automatize Framework for IASA (A3.3)." As detailed previously with the IDEF0 diagram of the function "Provide Information Assurance SA (A3)" description, there are several external inputs, outputs, constraints, and enablers that interact with A3.3.

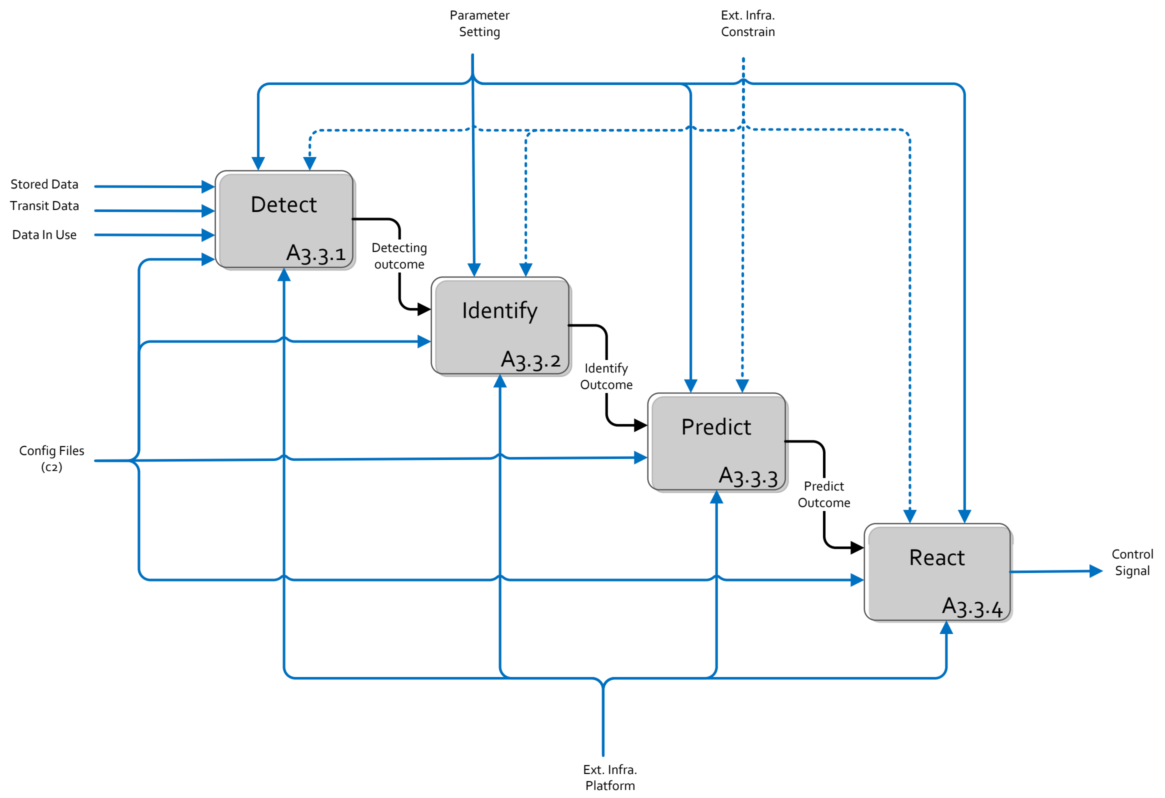


Figure 36. Level 3 IDEF0 diagram of function "Provide Intelligence Automatize Framework for IASA."

This diagram specifies with which level 3 functions of A3.3 the external inputs, outputs, constraints, and enablers interact. Interaction is defined as the transfer of EMMI to the external systems, as indicated in blue.

Inputs in the form of Stored Data (Data at Rest), Transit Data (Data in Motion), and Data In Use from A3.2 will input to the "Detect (A3.3.1)" function for analysis and detection of abnormality. Configuration files created by the A1 function will serve as inputs to sub-functions of A3.3 and determine their operation environment.

The Parameter Setting constraint from the A3.1 function is received by all sub-functions (A3.3.1 to A3.3.4) of A3.3 and will set how the sub-functions operate. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all four functions (A3.3.1 to A3.3.4).

From the data gathered from the inputs, after going through the DIPR analysis, the required control signal is created as output to “React with Alerts & Control Signal (A3.4)” for reaction to the situation.

The enabler, External Infrastructure Platform, will determine how all the functions will operate in the cyber environment.

This diagram also specifies how the level 3 functions of A3.3 interact with each other. Interaction or the transfer of EMMI within the system/level is indicated in black. Data collected are analyzed and abnormalities are detected by the “Detect (A3.3.1)” function; the output from it is input to the “Identify (A3.3.2)” function. Function A3.3.2 further determines and identifies the possible causes and is output to the “Predict (A3.3.3)” function. Function A3.3.3 predicts the possible outcomes and outputs the analysis to the “React (A3.3.4)” function. Function A3.3.4 uses predetermined settings to select the reaction to the situation and outputs this selection as a control signal for the “React with Alerts & Control Signal (A3.4)” function.

6. Detect IDEF0 (A3.3.1 Level 4)

This level 4 IDEF0 diagram (Figure 37) depicts the functions of “Detect (A3.3.1).” As detailed previously with the IDEF0 diagram of the function “Provide Intelligence Automatize Framework for IASA (A3.3)” description, there are several external inputs, outputs, constraints, and enablers that interact with A3.3.1.

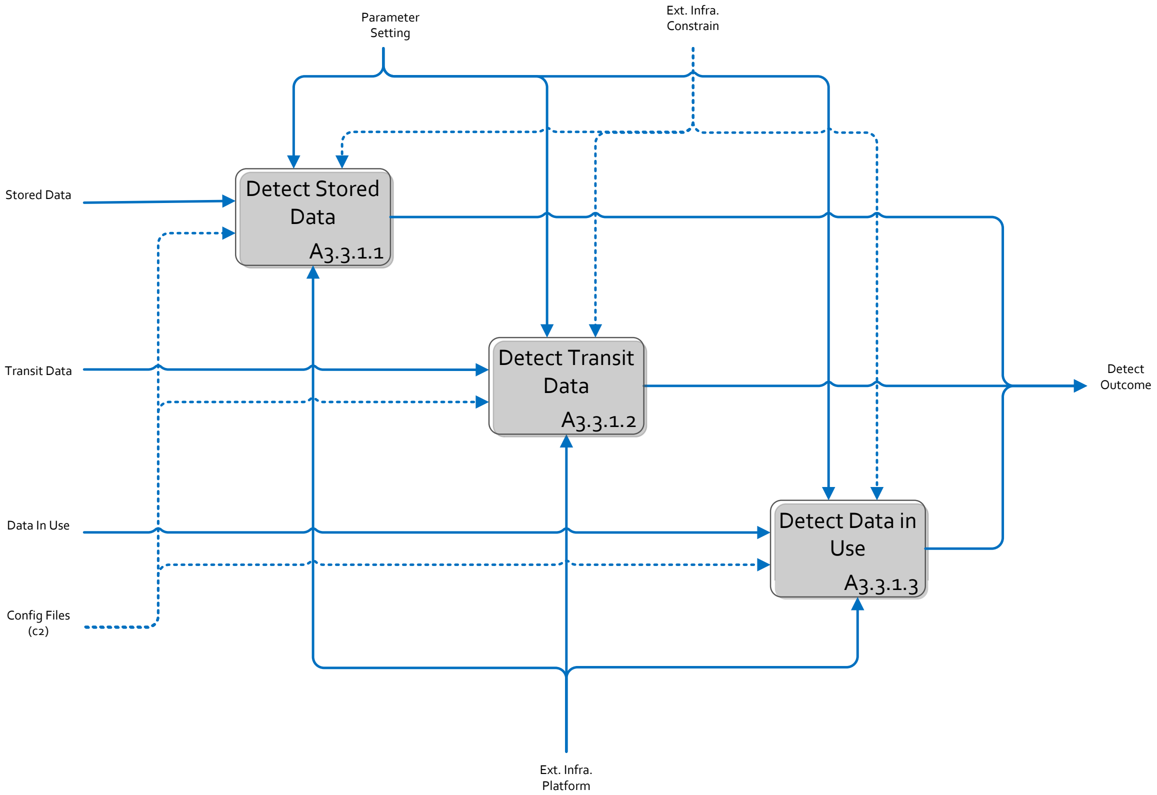


Figure 37. Level 4 IDEF0 diagram of function "Detect."

This diagram specifies with which level 4 functions of A3.3.1 the external inputs, outputs, constraints, and enablers interact. Interaction is defined as the transfer of EMMI with the external systems, as indicated in blue.

Depending on the types of Input (Stored Data, Transit Data and Data In Use), the sub-functions of A3.3.1 will receive them for detection function. Configuration files created by the A1 function will serve as inputs to sub-functions of A3.3.1 and determine their operation environment.

The Parameter Setting constraint from the A3.3 function is received by all sub-functions (A3.3.1.1 to A3.3.1.3) of A3.3.1 and will set how the sub-functions operate. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all three functions (A3.3.1.1 to A3.3.1.3).

After detection analysis, the detect outcome is output to “Identify (A3.3.2)” for identification against predetermined intelligent state rules.

The enabler, External Infrastructure Platform, will determine how all the functions will operate in the cyber environment.

There is no internal interaction between the sub-functions of A3.3.1.

7. Detect Stored/Transit/In Use Data IDEF0 (A3.3.1.X Level 5)

This level 5 IDEF0 diagram (Figure 38) depicts the functions of “Detect Stored/Transit/In Use Data (A3.3.1.X).” X represents the function numbering based on the type of data been processed (detection of stored data, data in transit, and in use) and is numbered 1, 2, and 3, respectively. As detailed previously with the IDEF0 diagram of the function “Detect (A3.3.1)” description, there are several external inputs, outputs, constraints, and enablers that interact with A3.3.1.X.

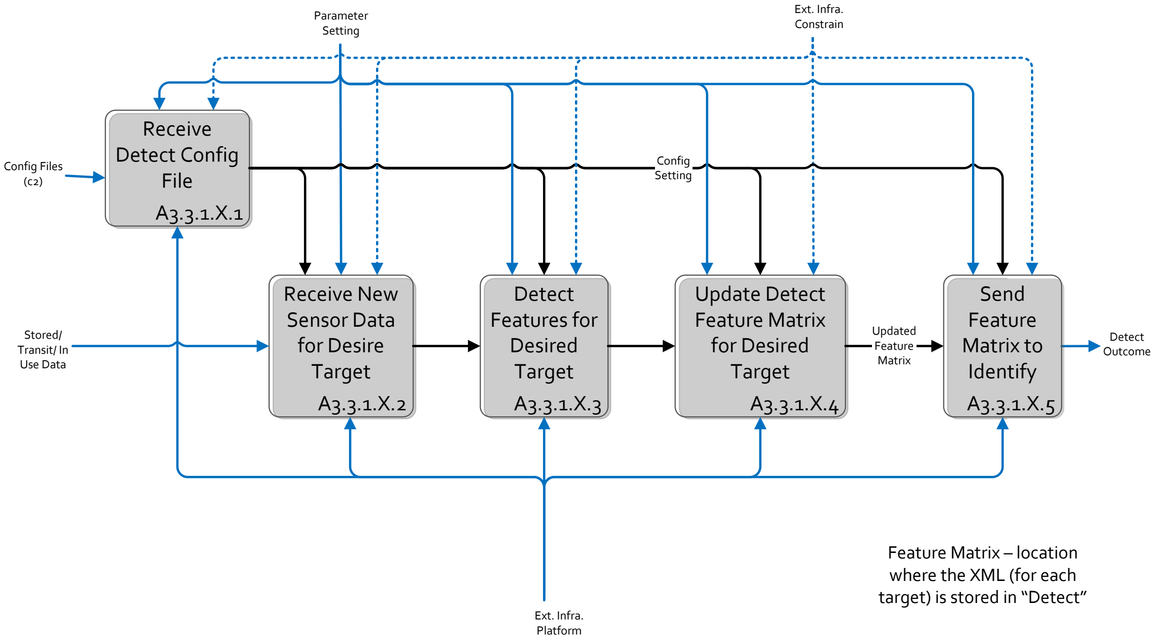

Parameter Setting
Ext. Infra. Constrain
Config Files (c2)
Receive Detect Config File
A3.3.1.X.1
Config Setting
Stored/ Transit/ In Use Data
Receive New Sensor Data for Desire Target
A3.3.1.X.2
Detect Features for Desired Target
A3.3.1.X.3
Update Detect Feature Matrix for Desired Target
A3.3.1.X.4
Updated Feature Matrix
Send Feature Matrix to Identify
A3.3.1.X.5
Detect Outcome
Feature Matrix – location where the XML (for each target) is stored in “Detect”
Ext. Infra. Platform

Figure 38. Level 5 IDEF0 diagram of function “Detect Stored/Transit/In Use Data.”

This diagram specifies with which level 5 functions of A3.3.1.X the external inputs, outputs, constraints, and enablers interact. Interaction is defined as the transfer of EMMI to the external systems, as indicated in blue.

Depending on the types of Input (Stored Data, Transit Data and Data In Use), the sub-functions of A3.3.1.X will receive them for the detection function (A3.3.1.X.2). Configuration files created by the A1 function will serve as inputs to function A3.3.1.X.1 which determines the operation environment of the other detection sub-functions (A3.3.1.X.2 to A3.3.1.X.5).

The Parameter Setting constraint from the A3.3 function is received by all sub-functions of A3.3.1.X and will set how the sub-functions operate. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all sub-functions of A3.3.1.X.

The updated detect feature matrix will be released by function A3.3.1.X.5 as the detection outcome sent to “Identify (A3.3.2)” for identification against predetermined intelligent state rules.

The enabler, External Infrastructure Platform, will determine how all the functions will operate and function in the cyber environment.

This diagram also specifies how the level 5 functions of A3.3.1.X interact with each other. Interaction or the transfer of EMMI within the system/level is indicated in black. “Receive Detect Config File (A3.3.1.X.1)” will provide the configuration setting as the parameter setting (constraint) to the other sub-functions of Detect (A3.3.1). After receiving the sensor data, “Received New Sensor Data for Desired Target (A3.3.1.X.2)” will output the data to function A3.3.1.X.3. Function A3.3.1.X.3 will sense and pick up the required information based on the given setting from the data collected from the smart clouds. Function A3.3.1.X.4 updates the detect feature matrix based on the information collected; the updated feature matrix is then passed on to function A3.3.1.X.5 for out-processing of the Detect (A3.3.1) function.

8. Identify IDEF0 (A3.3.2 Level 4)

This level 4 IDEF0 diagram (Figure 39) depicts the functions of "Identify (A3.3.2)." As described previously with the IDEF0 diagram of the function "Provide Intelligence Automatize Framework for IASA (A3.3)" description, there are several external inputs, outputs, constraints, and enablers that interact with A3.3.2.

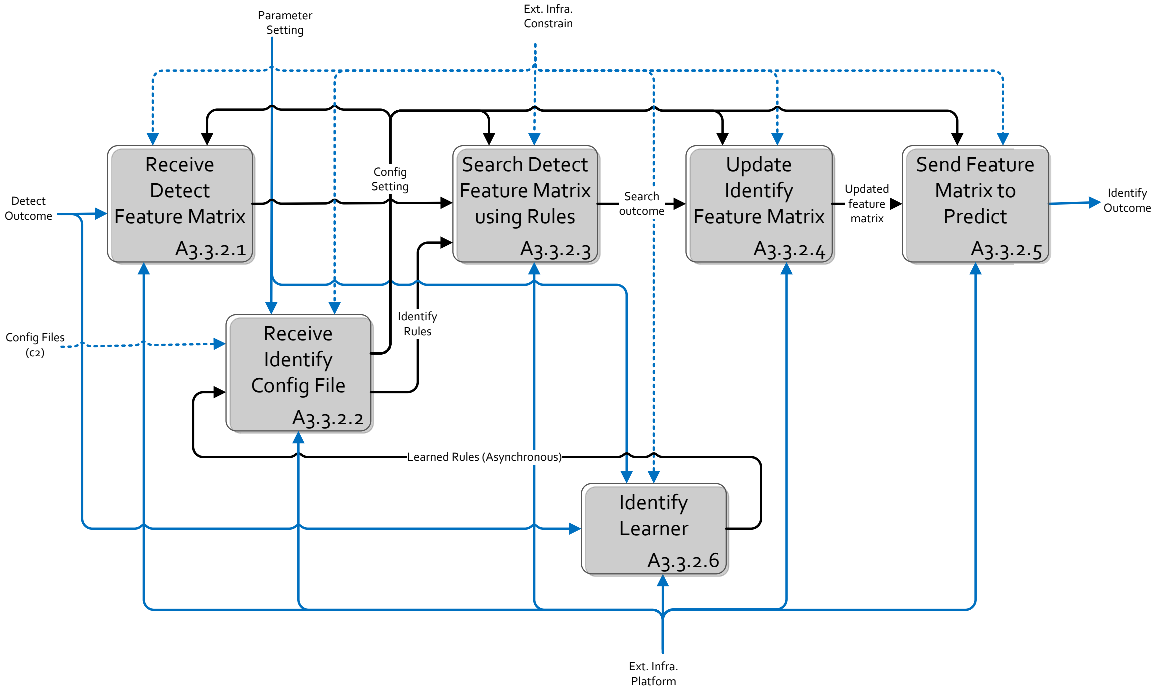


Figure 39. Level 4 IDEF0 diagram of function "Identify."

This diagram specifies with which level 4 functions of A3.3.2 the external inputs, outputs, constraints, and enablers interact. Interaction is defined as the transfer of EMMI to the external systems, as indicated in blue.

Inputs from the Detect Function will be used initially by "Identify Learner (A3.3.2.6)" to build up the intelligent state rules, subsequently servicing additional learned rules to "Receive Identify Config File (A3.3.2.2). Both functions operate asynchronously. Configuration files created by the A1 function will serve as inputs to sub-functions of A3.3.2.2, which determine the operation environment of the other identify sub-functions (A3.3.2.1, A3.3.2.3 to A3.3.2.5).

The Parameter Setting constraint from the A3.3.2 function is received by sub-functions A3.3.2.2 and A3.3.2.6 to determine how these functions operate and what intelligent state rules to implement. Intelligent state rules are rules that guide what the Identify function should look for in the feature matrix. The External Infrastructures constraint determines how the system functions interact and operate with the external systems and also with each other in The System. The External Infrastructures constraint goes to all four functions (A3.3.2.1 to A3.3.2.6).

After the identifying analysis is completed and the feature matrix is updated with the identification information by function A3.3.2.5, the outcome is output to "Predict (A3.3.3)" for prediction analysis.

The enabler, External Infrastructure Platform, will determine how all the functions will operate in the cyber environment.

This diagram also specifies how the level 4 functions of A3.3.2 interact with each other. Interaction or the transfer of EMMI within the system/level is indicated in black. After receiving the feature matrix from the detect function, "Receive Detect Feature Matrix (A3.3.2.1) will output the matrix to function "Search Detect Feature Matrix using Rules (A3.3.2.3)." The consolidated intelligent state rules from "Receive Identify Config File (A3.3.2.2)" function will be used by function A3.3.2.3 to perform the search function of the entire collected "Detect" feature matrix. Function A3.3.2.2 also provides the configuration setting as the parameter setting (constraint) to the other sub-function of Identify (A3.3.2). The search outcome from Function A3.3.2.3 will update the feature matrix with identification information by the "Update Identify Feature Matrix (A3.3.2.4)" function; the updated feature matrix is then passed on to function A3.3.2.5 for out-processing of Identify (A3.3.2) function. Function A3.3.2.6 will provide additional learned rules to function A3.3.2.2.

G. REQUIREMENTS ANALYSIS

This section itemizes the derived functional requirements corresponding to the IDEF0 level decompositions presented in the previous section.

1. High Level System (A0) Functional Requirements (Level 0)

Req. 0.1 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall receive user inputs from operator (B1).

Req. 0.2 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall receive monitoring targets from Smart Cloud SoS (B2).

Req. 0.3 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 0.4 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 0.5 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall adhere to policy constraint from Government Cyber Policy Departments (B3).

Req. 0.6 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall output alerts to operator (B1).

Req. 0.7 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall output cyber control signals to Smart Cloud SoS (B2).

Req. 0.8 Cyber Situational Awareness & Information Assurance of Smart Cloud SoS (A0) shall provide the function to obtain awareness of cyber situation and assurance of information of Smart Cloud SoS.

2. C2 (A1) Functional Requirements (Level 1)

Req. 1.1 C2 (A1) shall receive user inputs from operator (B1).

Req. 1.2 C2 (A1) shall adhere to policy constrain from Government Cyber Policy Department (B3).

Req. 1.3 C2 (A1) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 1.4 C2 (A1) shall output parameter setting to SA (A2), IASA (A3), and CADSA (A4).

Req. 1.5 C2 (A1) shall output configuration file to SA (A2), IASA (A3), and CADSA (A4).

Req. 1.6 C2 (A1) shall provide command and control function for the system.

3. SA (A2) Functional Requirements (Level 1)

Req. 2.1 SA (A2) shall receive configuration file from C2 (A1).

Req. 2.2 SA (A2) shall receive alert from IASA (A3) and CADSA (A4).

Req. 2.3 SA (A2) shall adhere to parameter setting constraint from C2 (A1).

Req. 2.4 SA (A2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 2.5 SA (A2) shall output filtered alert to operator (B1).

Req. 2.6 SA (A2) shall provide situational awareness function for the system.

4. IASA (A3) Functional Requirements (Level 1)

Req. 3.1 IASA (A3) shall receive configuration file from C2 (A1).

Req. 3.2 IASA (A3) shall receive monitoring targets from Smart Cloud SoS (B2).

Req. 3.3 IASA (A3) shall adhere to parameter setting constraint from C2 (A1).

Req. 3.4 IASA (A3) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.5 IASA (A3) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.6 IASA (A3) shall adhere to configuration/alert file from CADSA (A4).

Req. 3.7 IASA (A3) shall output cyber control signals to Smart Cloud SoS (B2).

Req. 3.8 IASA (A3) shall output configuration/alert file to CADSA (A4).

Req. 3.9 IASA (A3) shall output alert file to SA (A2).

Req. 3.10 IASA (A3) shall provide IASA function for the system.

5. CADSA (A4) Functional Requirements (Level 1)

Req. 4.1 CADSA (A4) shall receive configuration file from C2 (A1).

Req. 4.2 CADSA (A4) shall receive monitoring targets from Smart Cloud SoS (B2).

Req. 4.3 CADSA (A4) shall adhere to parameter setting constraint from C2 (A1).

Req. 4.4 CADSA (A4) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 4.5 CADSA (A4) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 4.6 CADSA (A4) shall adhere to configuration/alert file from IASA (A3).

Req. 4.7 CADSA (A4) shall output cyber control signals to Smart Cloud SoS (B2).

Req. 4.8 CADSA (A4) shall output configuration/alert file to IASA (A3).

Req. 4.9 CADSA (A4) shall output alert file to SA (A2).

Req. 4.10 CADSA (A4) shall provide cyber attack detection situational awareness function for the system.

6. Receive & Configure Files/Alerts (A3.1) Functional Requirements (Level 2)

Req. 3.1.1 Receive & Configure Files/Alerts (A3.1) shall receive configuration file from C2 (A1).

Req. 3.1.2 Receive & Configure Files/Alerts (A3.1) shall adhere to configuration/alert file from CADSA (A4).

Req. 3.1.3 Receive & Configure Files/Alerts (A3.1) shall adhere to parameter setting constraint from C2 (A1).

Req. 3.1.4 Receive & Configure Files/Alerts (A3.1) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.1.5 Receive & Configure Files/Alerts (A3.1) shall output parameter setting to Sense Information Data Environment (A3.2), Intelligence Automatize Framework for IASA (A3.3), and React with Alerts & Control Signal (A3.4).

Req. 3.1.6 Receive & Configure Files/Alerts (A3.1) shall provide receive and configure files/alerts function for IASA.

7. Sense Information Data Environment (A3.2) Functional Requirements (Level 2)

Req. 3.2.1 Sense Information Data Environment (A3.2) shall receive configuration file from C2 (A1).

Req. 3.2.2 Sense Information Data Environment (A3.2) shall receive monitoring targets from Smart Cloud SoS (B2).

Req. 3.2.3 Sense Information Data Environment (A3.2) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.2.4 Sense Information Data Environment (A3.2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.2.5 Sense Information Data Environment (A3.2) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.2.6 Sense Information Data Environment (A3.2) shall output feature matrix of Stored Data, Transit Data, and Data In Use collected from the monitored target.

Req. 3.2.7 Sense Information Data Environment (A3.2) shall provide the function to sense information data environment for the IASA.

8. Intelligence Automatize Framework for IASA (A3.3) Functional Requirements (Level 2)

Req. 3.3.1 Intelligence Automatize Framework for IASA (A3.3) shall receive feature matrix for Stored Data, Transit Data, and Data In Use from Sense Information Data Environment (A3.2).

Req. 3.3.2 Intelligence Automatize Framework for IASA (A3.3) shall receive configuration file from C2 (A1).

Req. 3.3.3 Intelligence Automatize Framework for IASA (A3.3) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.4 Intelligence Automatize Framework for IASA (A3.3) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.5 Intelligence Automatize Framework for IASA (A3.3) shall output control signal to React with Alerts & Control Signal (A3.4).

Req. 3.3.6 Intelligence Automatize Framework for IASA (A3.3) shall provide the function of intelligence automatize framework for IASA.

9. React with Alerts & Control Signal (A3.4) Functional Requirements (Level 2)

Req. 3.4.1 React with Alerts & Control Signal (A3.4) shall receive control signal from Intelligence Automatize Framework for IASA (A3.3).

Req. 3.4.2 React with Alerts & Control Signal (A3.4) shall receive configuration file from C2 (A1).

Req. 3.4.3 React with Alerts & Control Signal (A3.4) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.4.4 React with Alerts & Control Signal (A3.4) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.4.5 React with Alerts & Control Signal (A3.4) shall output cyber control signal to Smart Cloud SoS (B2).

Req. 3.4.6 React with Alerts & Control Signal (A3.4) shall output alert to SA (A2).

Req. 3.4.7 React with Alerts & Control Signal (A3.4) shall output configuration/alert file to CADSA (A4).

Req. 3.4.8 React with Alerts & Control Signal (A3.4) shall provide the function to react with alerts and control signals for IASA.

10. Sense Stored/Transmitted/In Use Information (A3.2.X) Functional Requirements (Level 3)

Letter X is used to represent the sub-functions of A3.2, covering Sense Stored Information (A3.2.1), Transmitted Information (A3.2.2) and In Use Information (A3.2.3).

Req. 3.2.X.1 Sense Stored/Transmitted/In Use Information (A3.2.X) shall receive configuration file from C2 (A1).

Req. 3.2.X.2 Sense Stored/Transmitted/In Use Information (A3.2.X) shall receive monitoring targets from Smart Cloud SoS (B2)

Req. 3.2.X.3 Sense Stored/Transmitted/In Use Information (A3.2.X) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.2.X.4 Sense Stored/Transmitted/In Use Information (A3.2.X) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.2.X.5 Sense Stored/Transmitted/In Use Information (A3.2.X) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.2.X.6 Sense Stored/Transmitted/In Use Information (A3.2.X) shall output feature matrix of Stored/Transmitted/In Use data collected from the monitored target.

*Req. 3.2.X.*7 Sense Stored/Transmitted/In Use Information (A3.2.X) shall provide the function to sense Stored/Transmitted/In Use information for intelligence automatize framework for IASA.

11. Receive/Process Config File (A3.2.X.1) Functional Requirements (Level 4)

Letter X is used to represent the sub-functions of A3.2, covering Sense Stored Information (A3.2.1), Transmitted Information (A3.2.2) and In Use Information (A3.2.3).

Req. 3.2.X.1.1 Receive/Process Config File (A3.2.X.1) shall receive monitoring targets from Smart Cloud SoS (B2).

Req. 3.2.X.1.2 Receive/Process Config File (A3.2.X.1) shall receive configuration file from C2 (A1).

Req. 3.2.X.1.3 Receive/Process Config File (A3.2.X.1) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.2.X.1.4 Receive/Process Config File (A3.2.X.1) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.2.X.1.5 Receive/Process Config File (A3.2.X.1) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.2.X.1.6 Receive/Process Config File (A3.2.X.1) shall output processed configuration file to Sense Environment for Desired Target (A3.2.X.2).

Req. 3.2.X.1.7 Receive/Process Config File (A3.2.X.1) shall provide the function to receive and process configuration file for sense data information function of IASA.

12. Sense Environment for Desired Target (A3.2.X.2) Functional Requirements (Level 4)

Letter X is used to represent the sub-functions of A3.2, covering Sense Stored Information (A3.2.1), Transmitted Information (A3.2.2) and In Use Information (A3.2.3).

Req. 3.2.X.2.1 Sense Environment for Desired Target (A3.2.X.2) shall receive monitoring targets from Receive/Process Config File (A3.2.X.1).

Req. 3.2.X.2.2 Sense Environment for Desired Target (A3.2.X.2) shall receive configuration file from C2 (A1).

Req. 3.2.X.2.3 Sense Environment for Desired Target (A3.2.X.2) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.2.X.2.4 Sense Environment for Desired Target (A3.2.X.2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.2.X.2.5 Sense Environment for Desired Target (A3.2.X.2) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.2.X.2.6 Sense Environment for Desired Target (A3.2.X.2) shall output sensed target data to Write/Record Sensor Result (A3.2.X.3).

Req. 3.2.X.2.7 Sense Environment for Desired Target (A3.2.X.2) shall provide the function to sense data from monitored targets for sense data information function of IASA.

13. Write/Record Sensor Result (A3.2.X.3) Functional Requirements (Level 4)

Letter X is used to represent the sub-functions of A3.2, covering Sense Stored Information (A3.2.1), Transmitted Information (A3.2.2) and In Use Information (A3.2.3).

Req. 3.2.X.3.1 Write/Record Sensor Result (A3.2.X.3) shall receive sensed target data from Sense Environment for Desired Target (A3.2.X.2).

Req. 3.2.X.3.2 Write/Record Sensor Result (A3.2.X.3) shall receive configuration file from C2 (A1).

Req. 3.2.X.3.3 Write/Record Sensor Result (A3.2.X.3) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.2.X.3.4 Write/Record Sensor Result (A3.2.X.3) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.2.X.3.5 Write/Record Sensor Result (A3.2.X.3) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.2.X.3.6 Write/Record Sensor Result (A3.2.X.3) shall output sensor result to Transmit Sensor Data (A3.2.X.4).

Req. 3.2.X.3.7 Write/Record Sensor Result (A3.2.X.3) shall provide the function to write and record sensor result for sense data information function of IASA.

14. Transmit Sensor Data (A3.2.X.4) Functional Requirements (Level 4)

Letter X is used to represent the sub-functions of A3.2, covering Sense Stored Information (A3.2.1), Transmitted Information (A3.2.2), and In Use Information (A3.2.3).

Req. 3.2.X.4.1 Transmit Sensor Data (A3.2.X.4) shall receive sensor result from Write/Record Sensor Result (A3.2.X.3).

Req. 3.2.X.4.2 Transmit Sensor Data (A3.2.X.4) shall receive configuration file from C2 (A1).

Req. 3.2.X.4.3 Transmit Sensor Data (A3.2.X.4) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.2.X.4.4 Transmit Sensor Data (A3.2.X.4) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.2.X.4.5 Transmit Sensor Data (A3.2.X.4) shall adhere to infrastructure constraint from Smart Cloud SoS (B2).

Req. 3.2.X.4.6 Transmit Sensor Data (A3.2.X.4) shall output sensor data to Detect (A3.3.1).

Req. 3.2.X.4.7 Transmit Sensor Data (A3.2.X.4) shall provide the function to transmit sensor data for sense data information function of IASA.

15. Detect (A3.3.1) Functional Requirements (Level 3)

Req. 3.3.1.1 Detect (A3.3.1) shall receive feature matrix for Stored Data, Transit Data, and Data In Use from (A3.2).

Req. 3.3.1.2 Detect (A3.3.1) shall receive configuration file from C2 (A1).

Req. 3.3.1.3 Detect (A3.3.1) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.4 Detect (A3.3.1) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.1.5 Detect (A3.3.1) shall output detection outcome to Identify (A3.3.2).

Req. 3.3.1.6 Detect (A3.3.1) shall provide the detection function for intelligence automatize framework for IASA.

16. Identify (A3.3.2) Functional Requirements (Level 3)

Req. 3.3.2.1 Identify (A3.3.2) shall receive detection outcome from Detect (A3.3.1).

Req. 3.3.2.2 Identify (A3.3.2) shall receive configuration file from C2 (A1).

Req. 3.3.2.3 Identify (A3.3.2) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.2.4 Identify (A3.3.2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.5 Identify (A3.3.2) shall output identification outcome to Predict (A3.3.3).

Req. 3.3.2.6 Identify (A3.3.2) shall provide the identify function for intelligence automatize framework for IASA.

17. Predict (A3.3.3) Functional Requirements (Level 3)

Req. 3.3.3.1 Predict (A3.3.3) shall receive identification outcome from Identify (A3.3.2).

Req. 3.3.3.2 Predict (A3.3.3) shall receive configuration file from C2 (A1).

Req. 3.3.3.3 Predict (A3.3.3) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.3.4 Predict (A3.3.3) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.3.5 Predict (A3.3.3) shall output predication outcome to React (A3.3.4).

Req. 3.3.3.6 Predict (A3.3.3) shall provide the prediction function for intelligence automatize framework for IASA.

18. React (A3.3.4) Functional Requirements (Level 3)

Req. 3.3.4.1 React (A3.3.4) shall receive predication outcome from Predict (A3.3.3).

Req. 3.3.4.2 React (A3.3.4) shall receive configuration file from C2 (A1).

Req. 3.3.4.3 React (A3.3.4) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.4.4 React (A3.3.4) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.4.5 React (A3.3.4) shall output control signal to React with Alerts & Control Signal (A3.4).

Req. 3.3.4.6 React (A3.3.4) shall provide the reaction function for intelligence automatize framework for IASA.

19. Detect Stored/Transit/In Use Data (A3.3.1.X) Functional Requirements (Level 4)

Letter X is used to represent the sub-functions of A3.3.1, covering Sense Stored Information (A3.3.1.1), Transmitted Information (A3.3.1.2), and In Use Information (A3.3.1.3).

Req. 3.3.1.X.1 Detect Stored/Transit/In Use Data (A3.3.1.X) shall receive feature matrix for Stored/Transit/In Use Data from Sense Information Data Environment (A3.2).

Req. 3.3.1.X.2 Detect Stored/Transit/In Use Data (A3.3.1.X) shall receive configuration file from C2 (A1).

Req. 3.3.1.X.3 Detect Stored/Transit/In Use Data (A3.3.1.X) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.X.4 Detect Stored/Transit/In Use Data (A3.3.1.X) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.1.X.5 Detect Stored/Transit/In Use Data (A3.3.1.X) shall output detection outcome to Identify (A3.3.2).

Req. 3.3.1.1.6 Detect Stored/Transit/In Use Data (A3.3.1.X) shall provide the detection function for stored data.

20. Receive Detect Config File (A3.3.1.X.1) Functional Requirements (Level 5)

Letter X is used to represent the sub-functions of A3.3.1, covering Sense Stored Information (A3.3.1.1), Transmitted Information (A3.3.1.2), and In Use Information (A3.3.1.3).

Req. 3.3.1.X.1.1 Receive Detect Config File (A3.3.1.X.1) shall receive configuration file from C2 (A1).

Req. 3.3.1.X.1.2 Receive Detect Config File (A3.3.1.X.1) shall adhere to parameter setting constrain from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.X.1.3 Receive Detect Config File (A3.3.1.X.1) shall adhere to infrastructure constrain from External Infrastructures (B4).

Req. 3.3.1.X.1.4 Receive Detect Config File (A3.3.1.X.1) shall output configuration file to other functions of Detect Stored/Transit/In Use Data (A3.3.1.X).

Req. 3.3.1.X.1.5 Receive Detect Config File (A3.3.1.X.1) shall provide the function to receive detect configuration file for detection function of IASA.

21. Receive New Sensor Data for Desired Target (A3.3.1.X.2) Functional Requirements (Level 5)

Letter X is used to represent the sub-functions of A3.3.1, covering Sense Stored Information (A3.3.1.1), Transmitted Information (A3.3.1.2), and In Use Information (A3.3.1.3).

Req. 3.3.1.X.2.1 Receive New Sensor Data for Desired Target (A3.3.1.X.2) shall receive stored/transit/in use data from Sense Information Data Environment (A3.2).

Req. 3.3.1.X.2.2 Receive New Sensor Data for Desired Target (A3.3.1.X.2) shall adhere to configuration file constraint from Receive Detect Config File (A3.3.1.X.1).

Req. 3.3.1.X.2.3 Receive New Sensor Data for Desired Target (A3.3.1.X.2) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.X.2.4 Receive New Sensor Data for Desired Target (A3.3.1.X.2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.1.X.2.5 Receive New Sensor Data for Desired Target (A3.3.1.X.2) shall output received sensor data to Detect Features for Desired Target (A3.3.1.X.3).

Req. 3.3.1.X.2.6 Receive New Sensor Data for Desired Target (A3.3.1.X.2) shall provide the function to receive new sensor data for detection function of IASA.

22. Detect Features for Desired Target (A3.3.1.X.3) Functional Requirements (Level 5)

Letter X is used to represent the sub-functions of A3.3.1, covering Sense Stored Information (A3.3.1.1), Transmitted Information (A3.3.1.2), and In Use Information (A3.3.1.3).

Req. 3.3.1.X.3.1 Detect Features for Desired Target (A3.3.1.X.3) shall receive sensor data from Receive New Sensor Data for Desired Target (A3.3.1.X.2).

Req. 3.3.1.X.3.2 Detect Features for Desired Target (A3.3.1.X.3) shall adhere to configuration file constraint from Receive Detect Config File (A3.3.1.X.1).

Req. 3.3.1.X.3.3 Detect Features for Desired Target (A3.3.1.X.3) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.X.3.4 Detect Features for Desired Target (A3.3.1.X.3) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.1.X.3.5 Detect Features for Desired Target (A3.3.1.X.3) shall output detected features to Update Detect Feature Matrix for Desired Target (A3.3.1.X.4).

Req. 3.3.1.X.3.6 Detect Features for Desired Target (A3.3.1.X.3) shall provide the function to detect features of targets for detection function of IASA.

23. Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) Functional Requirements (Level 5)

Letter X is used to represent the sub-functions of A3.3.1, covering Sense Stored Information (A3.3.1.1), Transmitted Information (A3.3.1.2), and In Use Information (A3.3.1.3).

Req. 3.3.1.X.4.1 Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) shall receive detect features from Detect Features for Desired Target (A3.3.1.X.3).

Req. 3.3.1.X.4.2 Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) shall adhere to configuration file constraint from Receive Detect Config File (A3.3.1.X.1).

Req. 3.3.1.X.4.3 Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.X.4.4 Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.1.X.4.5 Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) shall output the updated feature matrix to Send Feature Matrix to Identify (A3.3.1.X.5).

Req. 3.3.1.X.4.6 Update Detect Feature Matrix for Desired Target (A3.3.1.X.4) shall provide the function to update the feature matrix of targets for detection function of IASA.

24. Send Feature Matrix to Identify (A3.3.1.X.5) Functional Requirements (Level 5)

Letter X is used to represent the sub-functions of A3.3.1, covering Sense Stored Information (A3.3.1.1), Transmitted Information (A3.3.1.2), and In Use Information (A3.3.1.3).

Req. 3.3.1.X.5.1 Send Feature Matrix to Identify (A3.3.1.X.5) shall receive the updated feature matrix from Update Detect Feature Matrix for Desired Target (A3.3.1.X.4).

Req. 3.3.1.X.5.2 Send Feature Matrix to Identify (A3.3.1.X.5) shall adhere to configuration file constraint from Receive Detect Config File (A3.3.1.X.1).

Req. 3.3.1.X.5.3 Send Feature Matrix to Identify (A3.3.1.X.5) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.1.X.5.4 Send Feature Matrix to Identify (A3.3.1.X.5) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.1.X.5.5 Send Feature Matrix to Identify (A3.3.1.X.5) shall output the detection outcome to Identify (A3.3.2).

Req. 3.3.1.X.5.6 Send Feature Matrix to Identify (A3.3.1.X.5) shall provide the function to send the feature matrix of targets for detection function of IASA.

25. Receive Detect Feature Matrix (A3.3.2.1) Functional Requirements (Level 4)

Req. 3.3.2.1.1 Receive Detect Feature Matrix (A3.3.2.1) shall receive the detection outcome from Detect (A3.3.1).

Req. 3.3.2.1.2 Receive Detect Feature Matrix (A3.3.2.2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.1.3 Receive Detect Feature Matrix (A3.3.2.2) shall adhere to configuration file constraint from Receive Identify Config Files (A3.3.2.2).

Req. 3.3.2.1.4 Receive Detect Feature Matrix (A3.3.2.2) shall output the received detection feature matrix to Search Detect Feature Matrix using Rules (A3.3.2.3).

Req. 3.3.2.1.5 Receive Detect Feature Matrix (A3.3.2.2) shall provide the function to receive detection outcome for the identify function of IASA.

26. Receive Identify Config File (A3.3.2.2) Functional Requirements (Level 4)

Req. 3.3.2.2.1 Receive Identify Config File (A3.3.2.2) shall receive learned rules from Identify Learner (A3.3.2.6).

Req. 3.3.2.2.2 Receive Identify Config File (A3.3.2.2) shall receive configuration file from C2 (A1).

Req. 3.3.2.2.3 Receive Identify Config File (A3.3.2.2) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.2.2.4 Receive Identify Config File (A3.3.2.2) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.2.5 Receive Identify Config File (A3.3.2.2) shall output configuration file to identify sub-functions that handle feature matrix.

Req. 3.3.2.2.6 Receive Identify Config File (A3.3.2.2) shall output identify rules to Search Detect Feature Matrix using Rules (A3.3.2.3).

Req. 3.3.2.2.7 Receive Identify Config File (A3.3.2.2) shall provide the function to receive configuration files and output them to sub-function of identify for the identify function of IASA.

27. Search Detect Feature Matrix using Rules (A3.3.2.3) Functional Requirements (Level 4)

Req. 3.3.2.3.1 Search Detect Feature Matrix using Rules (A3.3.2.3) shall receive the detection outcome from Receive Detect Feature Matrix (A3.3.2.1).

Req. 3.3.2.3.2 Search Detect Feature Matrix using Rules (A3.3.2.3) shall receive identify rules from Receive Intelligent State Rules (A3.3.2.2).

Req. 3.3.2.3.3 Search Detect Feature Matrix using Rules (A3.3.2.3) shall adhere to configuration file constraint from Receive Identify Config Files (A3.3.2.2).

Req. 3.3.2.3.4 Search Detect Feature Matrix using Rules (A3.3.2.3) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.3.5 Search Detect Feature Matrix using Rules (A3.3.2.3) shall output the search outcome to Update Identify Feature Matrix (A3.3.2.4).

Req. 3.3.2.3.6 Search Detect Feature Matrix using Rules (A3.3.2.3) shall provide the function to search the detect feature matrix using rules for the identify function of IASA.

28. Update Identify Feature Matrix (A3.3.2.4) Functional Requirements (Level 4)

Req. 3.3.2.4.1 Update Identify Feature Matrix (A3.3.2.4) shall receive search outcome from Search Detect Feature Matrix using Rules (A3.3.2.3).

Req. 3.3.2.4.2 Update Identify Feature Matrix (A3.3.2.4) shall adhere to configuration file constraint from Receive Identify Config Files (A3.3.2.2).

Req. 3.3.2.4.3 Update Identify Feature Matrix (A3.3.2.4) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.4.4 Update Identify Feature Matrix (A3.3.2.4) shall output the updated feature matrix to Send Feature Matrix to Predict (A3.3.2.5).

Req. 3.3.2.4.5 Update Identify Feature Matrix (A3.3.2.4) shall provide the function to update the identify feature matrix for the identify function of IASA.

29. Send Feature Matrix to Predict (A3.3.2.5) Functional Requirements (Level 4)

Req. 3.3.2.5.1 Send Feature Matrix to Predict (A3.3.2.5) shall receive the updated identify feature matrix from Update Identify Feature Matrix (A3.3.2.4).

Req. 3.3.2.5.2 Send Feature Matrix to Predict (A3.3.2.5) shall adhere to configuration file constraint from Receive Identify Config Files (A3.3.2.2).

Req. 3.3.2.5.3 Send Feature Matrix to Predict (A3.3.2.5) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.5.4 Send Feature Matrix to Predict (A3.3.2.5) shall output the identify outcome to Predict (A3.3.3).

Req. 3.3.2.5.5 Send Feature Matrix to Predict (A3.3.2.5) shall provide the function to send the identify outcome for the identify function of IASA.

30. Identify Learner (A3.3.2.6) Functional Requirements (Level 4)

Req. 3.3.2.6.1 Identify Learner (A3.3.2.6) shall receive detection outcome from Detect (A3.3.1).

Req. 3.3.2.6.2 Identify Learner (A3.3.2.6) shall adhere to parameter setting constraint from Receive & Configure Files/Alerts (A3.1).

Req. 3.3.2.6.3 Identify Learner (A3.3.2.6) shall adhere to infrastructure constraint from External Infrastructures (B4).

Req. 3.3.2.6.4 Identify Learner (A3.3.2.6) shall output learned rules to Receive Identify Config File (A3.3.2.2).

Req. 3.3.2.6.5 Identify Learner (A3.3.2.6) shall provide the function to learn the identify rules for the identify function of IASA.

H. PHYSICAL ARCHITECTURE OF IASA SYSTEM

Recall the overall proposed system from Chapter I consists of three subsystems and the main focus will be on IASA system (Figure 40).

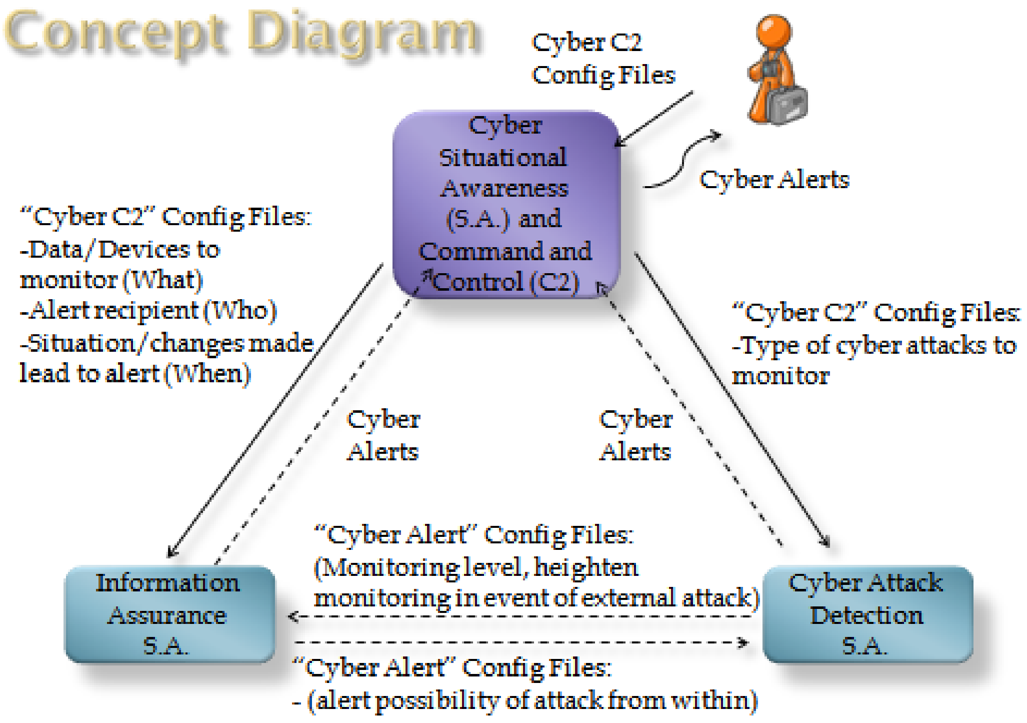


Figure 40. Focus on IASA in the cyber situational awareness system of systems of specified information in the cloud.

The deployment location of the DIPR software depends on the constraints caused by two factors. These are the bandwidths of the external infrastructure and the SoS computing resources and required performance. The following diagrams do not show the interactions between the IASA software modules; this will be discussed in Section G.

A smart cloud SoS with limited computing resources and high performance requirement will not want to handle the detection within its infrastructure; instead the sensor data are transferred across the network to the computer housing the DIPR software for analysis. This setup will free up valuable computing resources required by the smart cloud SoS to maintain its performance requirements; however, the network will be tasked to handle the transfer of the sensor data, and this will affect its available bandwidth. The physical architecture of such a setup is shown in Figure 41.

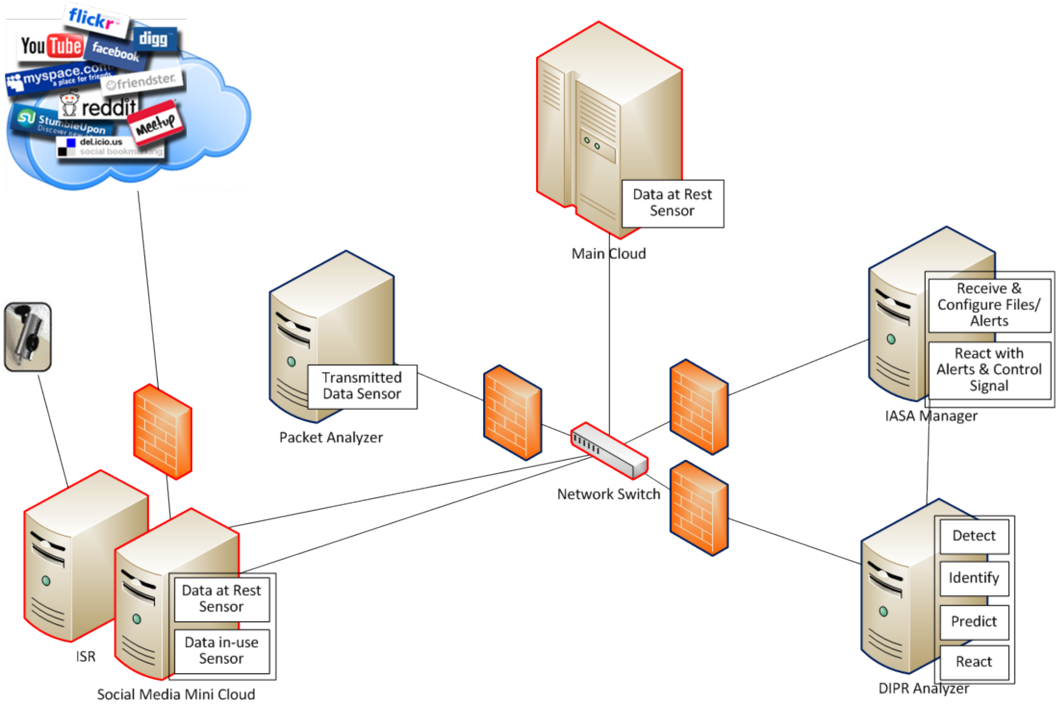


Figure 41. Physical Architecture of IASA with "Detect" function retained with IASA.

Network or computing equipment/resources that are highlighted in red are elements of the external systems interacting with IASA; these include the smart cloud (ISR and social media), firewall, and the hardware infrastructure supporting the system. Those elements highlighted in blue refer to the components of IASA which include the data sensor, the internal firewall, and DIPR software. Sensors for Data at Rest (stored data) are deployed within the locations where the targeted files are stored, such as within the main cloud or a mini cloud hard disk drive. Mini clouds are constantly updating their collected data (Data In Use) from social media, the Internet, and ISR; therefore, the in-use data sensors will be deployed within the mini clouds to closely monitor the targeted files. Some data will be transmitted from the mini clouds to the main cloud for processing or storage through the enterprise network. Packet analyzers with transmitted data sensors will be connected to the network to monitor the packets' delivery of the data. The DIPR analyzer will be running from the IASA system that is connected to the network; it will come with firewalls to prevent any unauthorized access to the IASA system from within the enterprise network.

SoS with limited bandwidth provided by the external infrastructure will require the detection software to be deployed at the Smart Cloud SoS. This setup will reduce the amount of filtered sensor data being transferred within the network to the computer that houses the DIPR analysis software, thereby freeing up valuable computing resources. With the detection done in the smart cloud infrastructure, only the feature matrix of the target data is transferred across the network to the computer housing the software for Identify, Predict, and React analysis. The physical architecture of such a setup is shown in Figure 42. The physical architecture of the system in this thesis will focus on assuming that the bandwidth of the network is limited and there are adequate computing resources in the smart cloud, as illustrated in Figure 42.

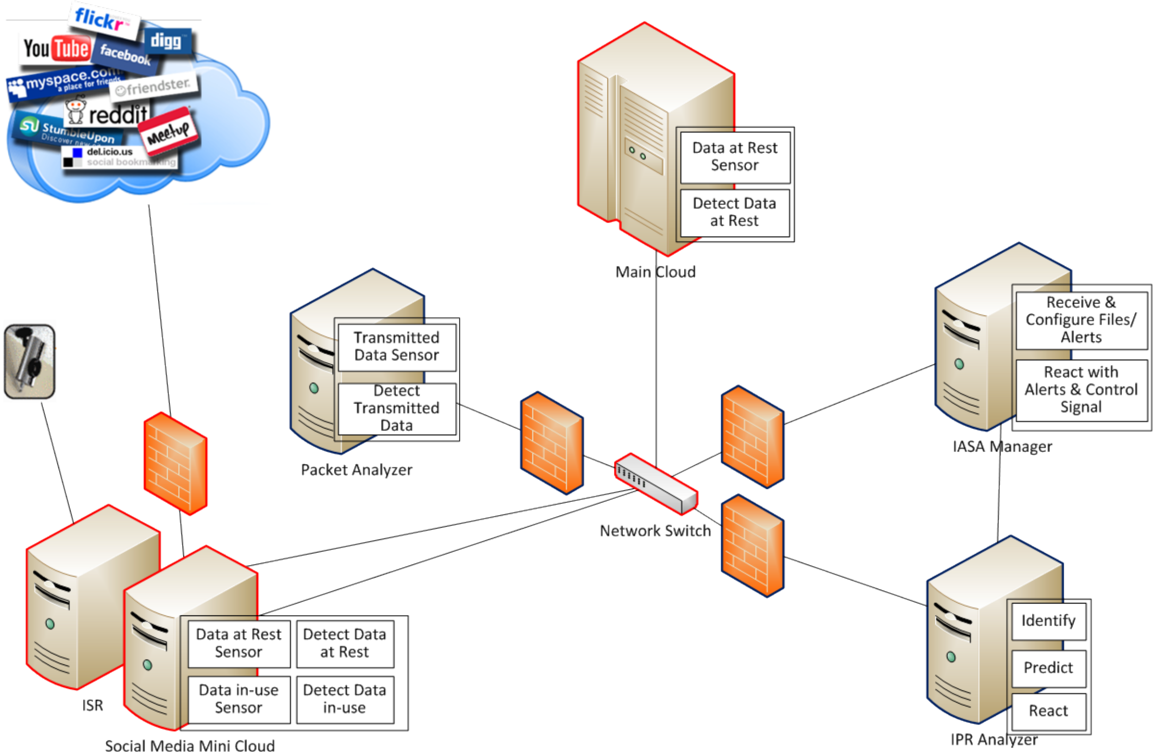


Figure 42. Physical Architecture of IASA with “Detect” function deployed at smart mini cloud SoS.

Figure 43 shows the flows of the configuration files to the sub-system of IASA in the physical architecture after receiving them from the C2 system. The IASA manager through “Receive & configure files/alerts” will receive configuration files from the C2 system and will pass the relevant configuration setting to the sub-system of IASA. All components of IASA, sense data software, and DIPR software, will receive the configuration files and adjust to match the monitoring needs set by the operator.

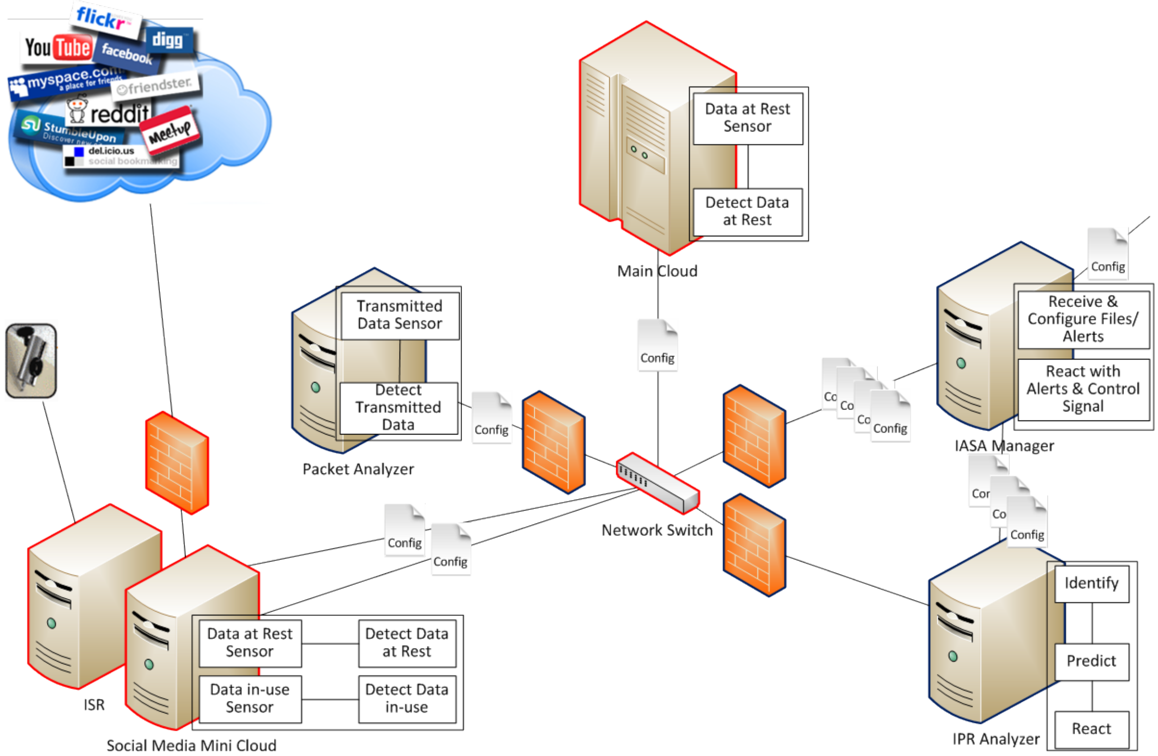


Figure 43. IASA receiving configuration files from C2 and filtering down to sub-systems.

Twitter JSON files and surveillance video files that have been collected from social media, the Internet, and cameras, respectively, are been processed and stored in the social media and ISR mini cloud. The main cloud will also maintain a library of data collected over a period of time. The stored data sensor monitors and reports the status of the saved files in the hard disk drive of the mini clouds and main cloud in the form of a text document. The Data In Use sensor will monitor files that are being processed in the memory of the mini cloud and will report the status in the form of a text document. Some of these data, in the form of an XML file, collected by the mini clouds will be transferred to the main cloud via the internal enterprise network. The transmitted data sensor deployed in the packet analyzer will monitor and report the status of the transit data in the form of an XML file. The IASA detection software for different states of data will pick up these XML files for analysis (Figure 44).

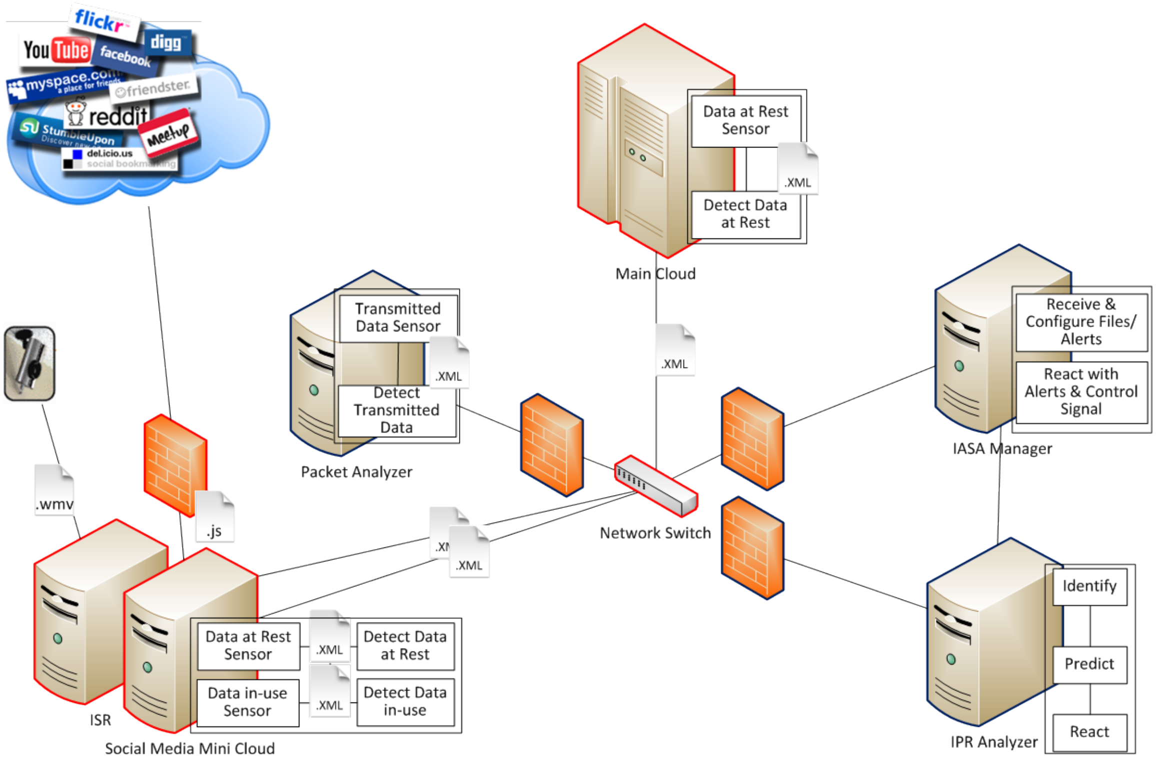


Figure 44. Data sensor monitoring targets' information.

The IASA Detect software will create feature matrix for each individual monitored file in the form of an XML file. XML files created by the detection software deployed in the mini clouds, main cloud, and packet analyzer will deliver the XML files through the enterprise network to the Identify, Predict, and React (IPR) analyzer for Identify analysis (Figure 45).

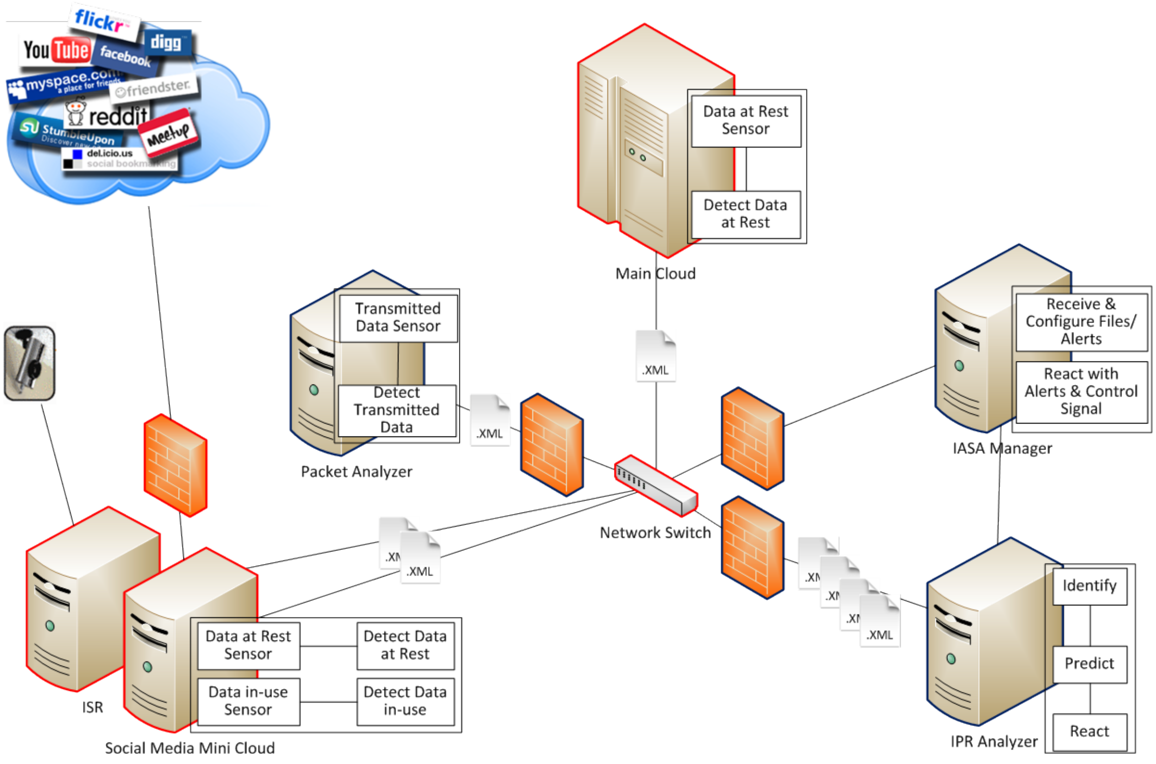


Figure 45. Detect processing sensor data and updating feature matrix.

The IASA Identify software receives the feature matrix XML files and compares them against predetermined intelligent state rules and updates the feature matrix XML file with information regarding any changes in the state rules. The Predict software using the identity outcome, calculates the possibility of an event occurring and updates the feature matrix XML file. The React software analyzes the probability of the event and responds with the appropriate measures to the IASA manager. Configuration files, alerts, and control signals can be issued by the IASA manager through “React with Alerts & Control Signal” in the form of an XML file. The Keyhole Markup Language (KML) file will be used as an alert when the location of the occurrence is of significant interest to the operator (Figure 46).

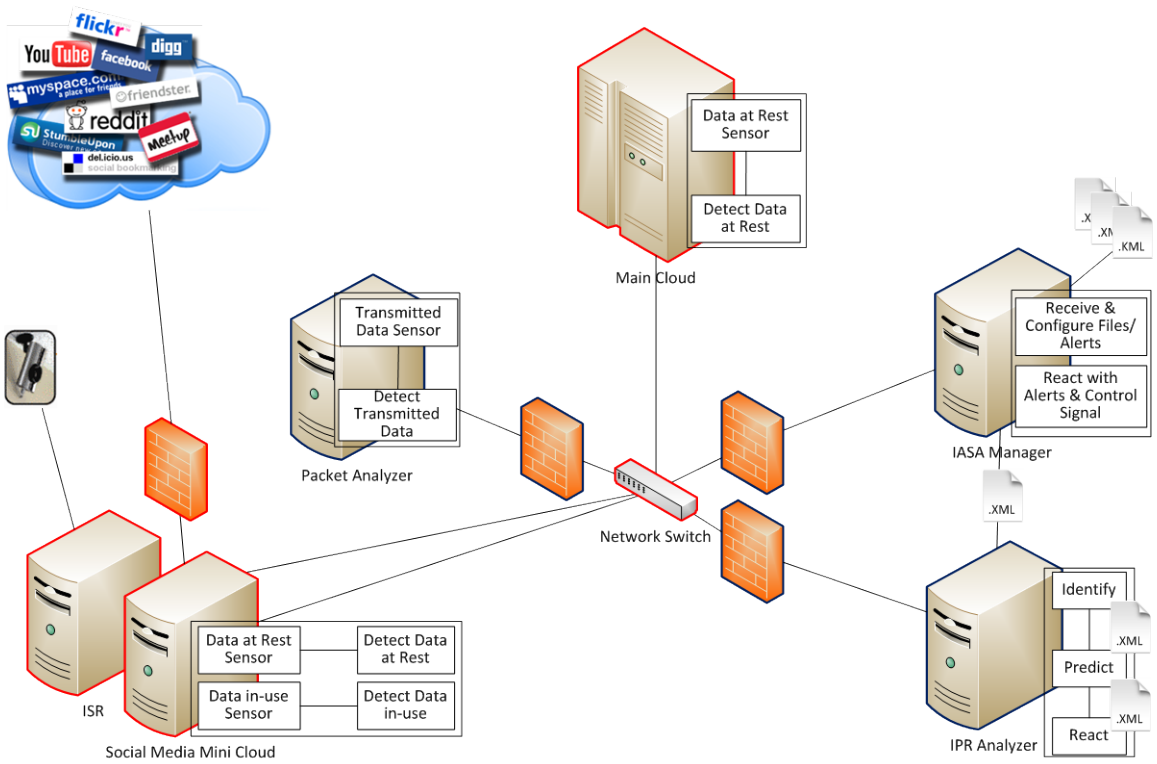


Figure 46. Identify, predict, and react software analyzing and reacting to the collected data.

I. SOFTWARE ARCHITECTURE

The software architecture of the IASA system can be described in Figure 47 with the assumption that the smart cloud SoS consists of mainly social media mini clouds and ISR mini clouds with a central database, called the main cloud. In the diagram, the social media mini cloud is represented in blue and the ISR mini cloud in red.

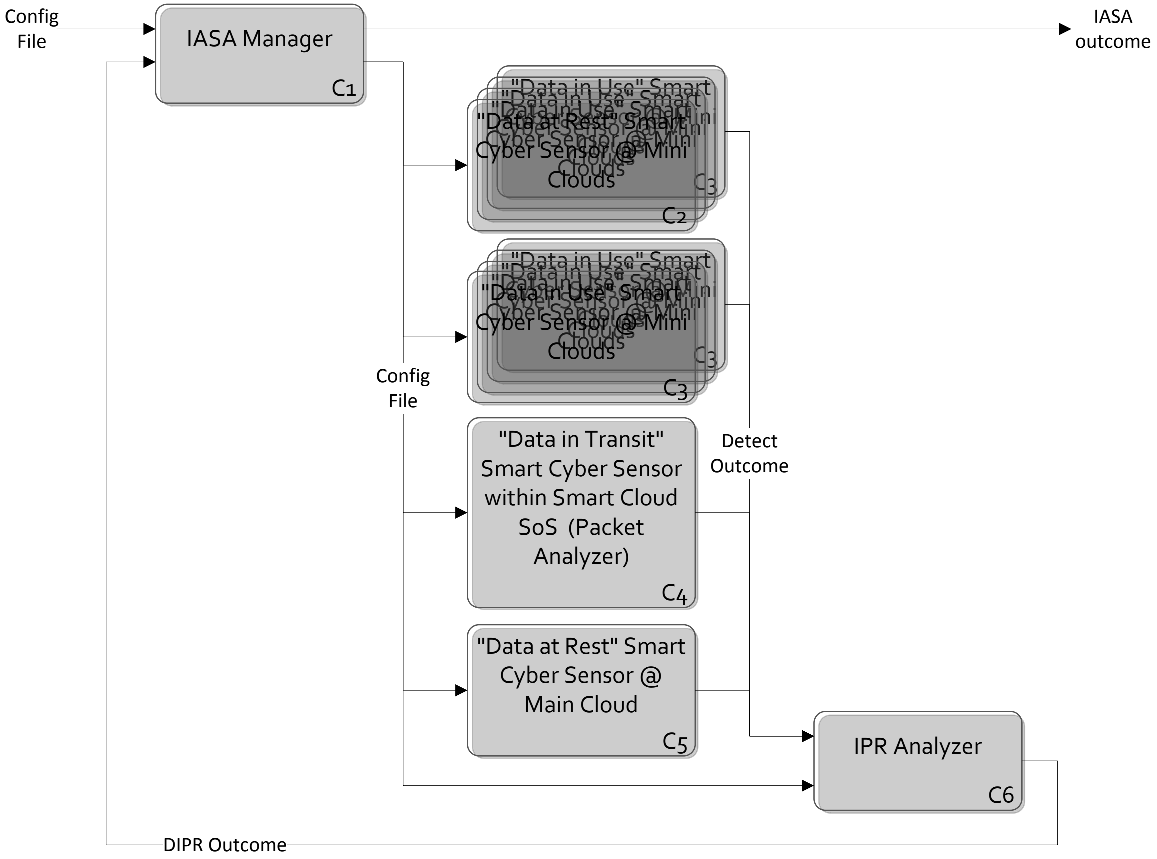

Config File
IASA Manager
C1
IASA outcome
"Data in Use" Smart Cyber Sensor @ Mini Clouds
C3
"Data at Rest" Smart Cyber Sensor @ Mini Clouds
C2
"Data in Use" Smart Cyber Sensor @ Mini Clouds
C3
Config File
"Data in Transit" Smart Cyber Sensor within Smart Cloud SoS (Packet Analyzer)
C4
Detect Outcome
"Data at Rest" Smart Cyber Sensor @ Main Cloud
C5
IPR Analyzer
C6
DIPR Outcome

Figure 47. Software architecture of IASA

The IASA manager consists of two software modules called the “Receive & Configure Files/Alerts” and “React with Alert & Control Signal.” The “Receive & Configure Files/Alerts” will first receive and verify the configuration settings from the command and control system. Once the settings are accepted, the configuration setting will be issued to the respective smart cyber detection sensor modules deployed for different states of data. For Data at Rest, the detect smart cyber sensor modules will be deployed to all mini clouds, which include social media and ISR mini clouds, and the main smart cloud. For Data in Transit, the packet analyzer software module, with its own smart cyber detector, will be analyzing the data flow between the mini clouds and main smart cloud. For Data in Use, the detect smart cyber sensor modules will be deployed to only the mini clouds, which include social media and ISR mini clouds. After detection by the smart cyber sensor modules, the detect outcome is collected and consolidated by the IPR analyzer software module. Within the IPR analyzer module, the IPR module will analyze the detection outcome and output the DIPR result to the IASA manager. The “React with Alert & Control Signal” module will issue alerts and/or control signal to the CSAC2 and CADSA system.

J. TRACEABILITY MATRIX OF FUNCTIONAL ALLOCATION

The mapping of the physical architecture to the functions of the IASA system is shown in the traceability matrix (Table 2).

Physical Architecture of IASA

Functions of IASA

	IASA Manager, C1	Data at Rest smart cyber sensor @ mini cloud, C2	Data in Use smart cyber sensor @ mini cloud, C3	Data in Transit smart cyber sensor, C4	Data at Rest smart cyber sensor @ main cloud, C5	IPR Analyzer , C6
Receive & configure file/alerts, A3.1	✔					
Sense information data environment, A3.2		✔	✔	✔	✔	
Provide intelligence automatize framework, A3.3		✔	✔	✔	✔	✔
React with alerts & control signal, A3.4						✔

Table 2. Traceability matrix of the IASA system.

K. PROPOSED INTELLIGENCE AUTOMATION FEATURE MATRIX

The DIPR feature matrix allows analytics assessment of the sensor inputs of raw data using software modules to increase data intelligence and automate reactions based on predetermined scenarios. It provides a quick timestamp of the targeted data status, providing users with cyber situation awareness of the information. The proposed XML schema of the feature matrix for the DIPR model is shown in Figure 48.

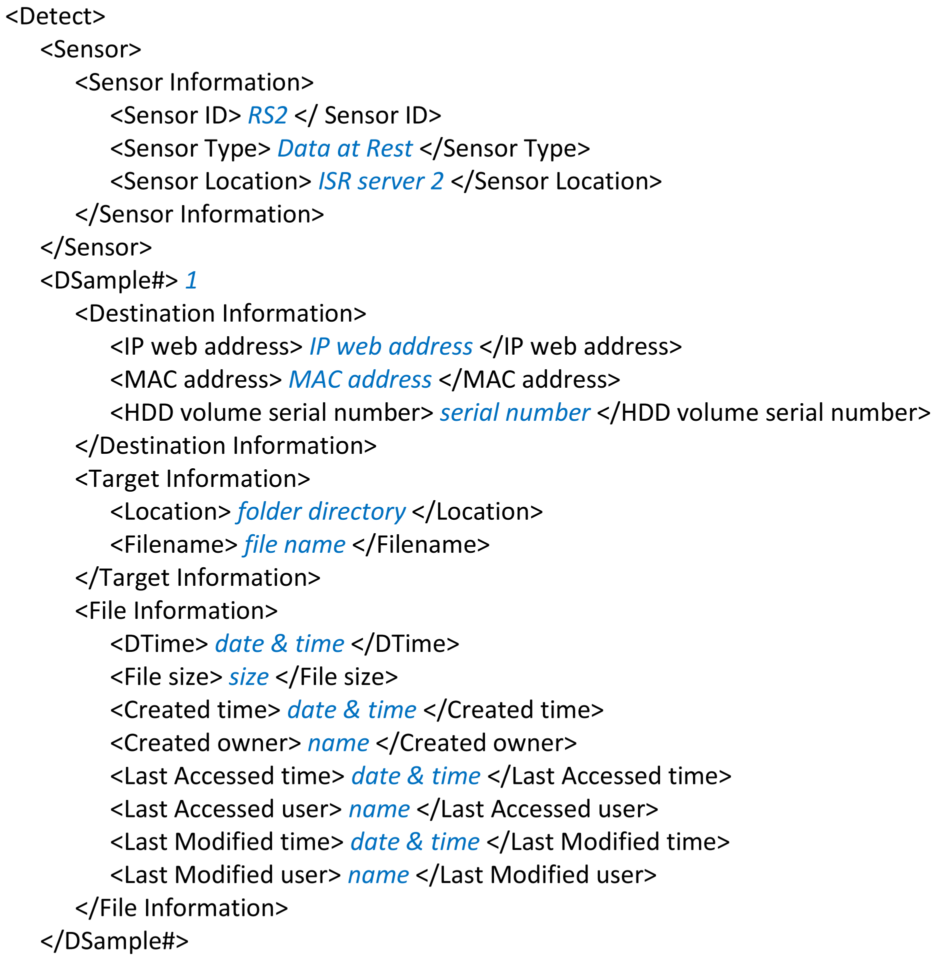
```
<?xml version="1.0"?>
<Detect>
    <Sensor>
        <Sensor Information>
            …
            …
        </Sensor Information>
    </Sensor>
    <DSample#> 1
        <Destination Information>
            …
            …
        </Destination Information>
        <Target Information>
            …
            …
        </Target Information>
        <File Information>
            …
            …
        </File Information>
    </DSample#>
    <DSample#> n
        <Destination Information>
            …
            …
        </Destination Information>
        <Target Information>
            …
            …
        </Target Information>
        <File Information>
            …
            …
        </File Information>
    </DSample#>
</Detect>
<Identify>
    <ISample#> 1
        …
        …
    </ISample#>
    <ISample#> n
        …
        …
    </ISample#>
</Identify>
<Predict>
    …
```

Figure 48. Overview of feature matrix XML schema.

The XML schema contains five main tags which represent the DIPR model. The first tag contains information regarding the smart cyber sensor that is used for detection. The "detect" tag contains information regarding the status of the data of interest collected by the sensors deployed in the clouds. Sampling rates of sensors can be fixed or of a configurable time period predetermined by the operator and set in the policy files. In the event of a heightened security situation, the sampling rate can be increased automatically. During detection by the function "Detect (A3.3.1)," the sensors will output XML files to the function "Receive New Sensor Data for Desired Target (A3.3.1.X.2)" and function "Detect Features for Desired Target (A3.3.1.X.3)" will pick up the required information pertaining to the data of interest. Information, such as where the data is located and saved, and the data itself will be picked up. This information will be recorded and updated on the feature matrix pertaining to the data by the function "Update Detect Feature Matrix for Desired Target (A3.3.1.X.4)." The updated feature matrix will then be sent out by "Send Feature Matrix to Identify (A3.3.1.X.5)."

The "identify" tag contains filtered information on the status of the data collected in the "detect" tag. It contains output of the function "Identify (A3.3.2)." Based on intelligent state rules, the state of the data will be recorded in the "identify" tag of the XML schema. Similar to detection, the feature matrix will first be received, searched, updated, and sent to the function "Predict (A3.3.3)." The "predict" tag contains the classifiers to recognize the behavior of the data, classifying it as normal, abnormal, or unclassified based on patterns. These behaviors will be used to predict the future state of the data under observation. The "predict" tag of the XML schema is updated by the function "Predict (A3.3.3). The "react" tag will provide an appropriate reaction to the change of the state of information, and this will serve as the control signal to be used by other systems in the SoS, such as to close certain ports or encrypt files or folders. The "react" tag is updated by the function "React (A3.3.4)."

The "Detect" sampling rate can be the same as sensors output rate but variable, depending on the configuration file sent to it. The policy for record keeping (Data Retention Policy) in the XML file can be fixed (e.g., 1 hr.). Sampling rates can vary depending on monitoring needs (e.g., 1 sec. to 1 min.), and the last record in the XML file which exceeds the time duration specified in the policy for record keeping will be overwritten with new data using a loop function (Figure 49).

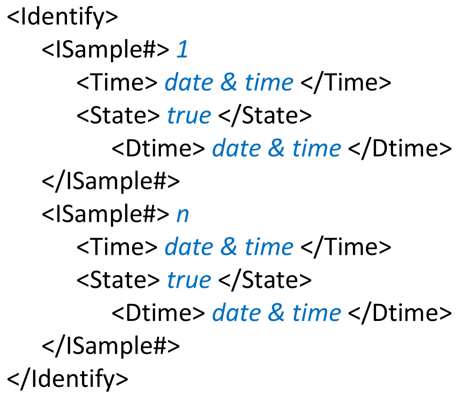


```
<Detect>
    <Sensor>
        <Sensor Information>
            <Sensor ID> RS2 </ Sensor ID>
            <Sensor Type> Data at Rest </Sensor Type>
            <Sensor Location> ISR server 2 </Sensor Location>
        </Sensor Information>
    </Sensor>
    <DSample#> 1
        <Destination Information>
            <IP web address> IP web address </IP web address>
            <MAC address> MAC address </MAC address>
            <HDD volume serial number> serial number </HDD volume serial number>
        </Destination Information>
        <Target Information>
            <Location> folder directory </Location>
            <Filename> file name </Filename>
        </Target Information>
        <File Information>
            <DTime> date & time </DTime>
            <File size> size </File size>
            <Created time> date & time </Created time>
            <Created owner> name </Created owner>
            <Last Accessed time> date & time </Last Accessed time>
            <Last Accessed user> name </Last Accessed user>
            <Last Modified time> date & time </Last Modified time>
            <Last Modified user> name </Last Modified user>
        </File Information>
    </DSample#>
```

Figure 49. “Detect” XML schema.

The “destination information” tag in the XML schema contains information about the host of the data such as the IP address of the device, the media access control (MAC) address, and the hard disk serial number. This information enables the hardware, where the data of interest is stored, to be monitored. The “target information” tag tracks the location or file directory where the data of interest is saved. The “file information” tag provides the status of the data of interest, such as time it was created, accessed and modified, and who performed those actions, and the size of the data. Together with target information, they provide a timestamp of the state of the data.

The “identify” tag (Figure 50) will only show the time when the sampling of the feature matrix shows a change in the state of the data. This process allows information to be intelligently filtered down to highlight the critical change in the state of the information and also to conserve computer resources, such as the size of the XML file. In each sampling tag, information regarding the state of data during sampling and the time the change of state was detected is recorded.



```
<Identify>
    <ISample#> 1
        <Time> date & time </Time>
        <State> true </State>
            <Dtime> date & time </Dtime>
    </ISample#>
    <ISample#> n
        <Time> date & time </Time>
        <State> true </State>
            <Dtime> date & time </Dtime>
    </ISample#>
</Identify>
```

Figure 50. “Identify” XML schema

L. PROPOSED INTELLIGENCE AUTOMATION CONFIGURATION FILE

The thesis will only focus only on sense, detect, and identify functions of the IASA system and the proposed XML schema for their configuration files is discussed in the following paragraphs.

1. Sensor Configuration File

The sensor configuration file contains seven main tags (Figure 51). Information regarding the operator issuing the configuration setting will be tagged under “originator/requestor.” Information pertaining to the sensor and target can be found under the “sensor information,” “target destination information,” and target information” tags, respectively. This information allows traceability on who issues the setting, what type of data is being monitored, and the physical location where the target is found. The configuration to obtain the status of the target will be provided in the “file information” tag. The option to activate encryption in the event of heightened security is also provided in the schema under the “encryption” tag. Finally, the location to output the sensor XML record is provided under the “output location” tag.

```
<?xml version="1.0"?>
<Sense configuration file>
    <Originator/Requestor>
        <C2 ID> C2 HQ </C2 ID>
        <User ID> User 1 </User ID>
        <C2time> date & time </C2time>
    </Originator/Requestor>
    <Sensor Information>
        <Sensor ID> RS2 </ Sensor ID>
        <Data Type> Data at Rest </Data Type>
        <Sensor> DIR </Sensor>
        <Sensor Location> ISR server 2 </Sensor Location>
        <Sensor Sampling Rate> 10sec </ Sensor Sampling Rate>
    </Sensor Information>
    <Target Destination Information>
        <TIP web address> IP web address </TIP web address>
        <TMAC address> MAC address </TMAC address>
        <THDD volume serial number> serial number </THDD volume serial number>
    </Target Destination Information>
    <Target Information>
        <TLocation> folder directory </Location>
        <TFilename> file name </Filename>
    </Target Information>
    <File Information>
        <DTime> Yes/No </DTime>
        <File size> Yes/No </File size>
        <Created time> Yes/No </Created time>
        <Created owner> Yes/No </Created owner>
        <Last Accessed time> Yes/No </Last Accessed time>
        <Last Accessed user> Yes/No </Last Accessed user>
        <Last Modified time> Yes/No </Last Modified time>
        <Last Modified user> Yes/No </Last Modified user>
    </File Information>
    <Encryption>
        <Encrypt Target> Yes/No </Encrypt Target>
    </ Encryption>
    <Output location>
        <OIP web address> IP web address </OIP web address>
        <OMAC address> MAC address </OMAC address>
        <OHDD volume serial number> serial number </OHDD volume serial number>
        <OLocation> folder directory </OLocation>
        <OFilename> file name </OFilename>
    </Output location>
</Sense configuration file>
```

Figure 51.   Sense configuration XML schema.

2. Detect Configuration File

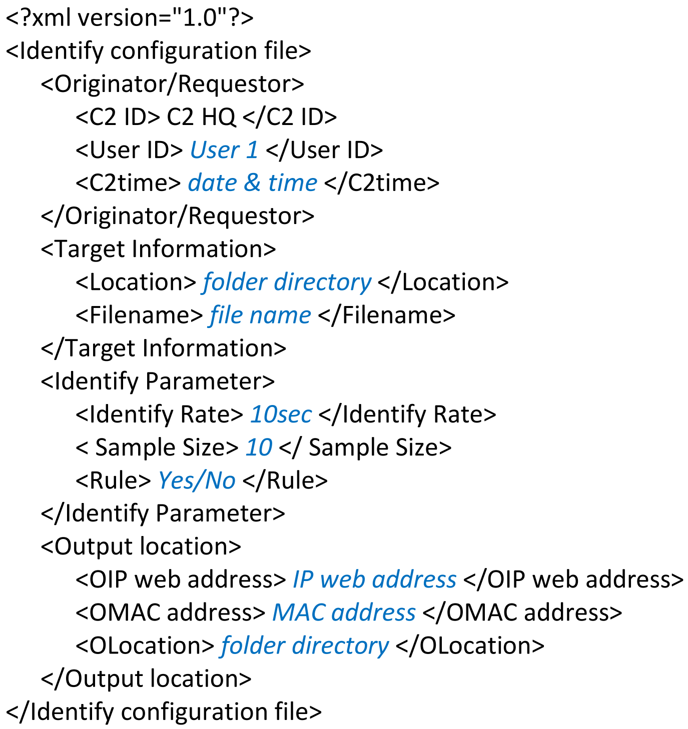
The detect configuration file contains five main tags (Figure 52). Information regarding the operator issuing the configuration setting is tagged under “originator/requestor.” Information about where the target can be found is provided under the “destination and target information” tag. This information allows traceability on who issues the setting and the physical location where the target is found. The configuration to set the detection parameter is provided under the “detect parameter” tag. The location of where to send the feature matrix after detection appears under the “output location” tag.

```
<?xml version="1.0"?>
<Detect configuration file>
   <Originator/Requestor>
      <C2 ID> C2 HQ </C2 ID>
      <User ID> User 1 </User ID>
      <C2time> date & time </C2time>
   </Originator/Requestor>
   <Destination Information>
      <IP web address> IP web address </IP web address>
      <MAC address> MAC address </MAC address>
      <HDD volume serial number> serial number </HDD volume serial number>
   </Destination Information>
   <Target Information>
      <Location> folder directory </Location>
      <Filename> file name </Filename>
   </Target Information>
   <Detect Parameter>
      <Detect Rate> 1sec </Detect Rate>
      <Sample Size> 10 </ Sample Size>
      <DTime> Yes/No </DTime>
      <File size> Yes/No </File size>
      <Created time> Yes/No </Created time>
      <Created owner> Yes/No </Created owner>
      <Last Accessed time> Yes/No </Last Accessed time>
      <Last Accessed user> Yes/No </Last Accessed user>
      <Last Modified time> Yes/No </Last Modified time>
      <Last Modified user> Yes/No </Last Modified user>
   </Detect Parameter>
   <Output location>
      <OIP web address> IP web address </OIP web address>
      <OMAC address> MAC address </OMAC address>
      <OLocation> folder directory </OLocation>
   </Output location>
</Detect configuration file>
```

Figure 52. Detect configuration XML schema.

3. Identify Configuration File

The identify configuration file contains four main tags (Figure 53). Information regarding the operator issuing the configuration setting will be tagged under “originator/requestor.” Information pertaining to the location where the target can be found is provided under the “target information” tag. This information allows traceability on who issues the setting and the location where the target is found. The configuration to set the identification sampling rate will be provided in the “identify parameter” tag. Finally, the location of where to output the feature matrix after identification is provided under the “output location” tag.



```
<?xml version="1.0"?>
<Identify configuration file>
   <Originator/Requestor>
      <C2 ID> C2 HQ </C2 ID>
      <User ID> User 1 </User ID>
      <C2time> date & time </C2time>
   </Originator/Requestor>
   <Target Information>
      <Location> folder directory </Location>
      <Filename> file name </Filename>
   </Target Information>
   <Identify Parameter>
      <Identify Rate> 10sec </Identify Rate>
      < Sample Size> 10 </ Sample Size>
      <Rule> Yes/No </Rule>
   </Identify Parameter>
   <Output location>
      <OIP web address> IP web address </OIP web address>
      <OMAC address> MAC address </OMAC address>
      <OLocation> folder directory </OLocation>
   </Output location>
</Identify configuration file>
```

Figure 53. Identify configuration XML schema.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. PROOF OF CONCEPT OF THE SYSTEM

A. SCOPE AND EXPERIMENT DESIGN

For scoping purposes, this proof of concept is scoped to monitor one specific feature matrix produced by the Detect software of IASA, which has filtered the data status of a target of interest (data at rest/stored data). In this case, the target of interest contains information (a tweet of a picture) that is collected from a social media mini cloud that is using the Windows operating system. The proof of concept will be using the Identify software of the DIPR model to analyze the feature matrix produced by the Detect software.

Our experiment assumes there is a smart cloud system of systems which has a social media mini cloud that senses social media. It also assumes users want to use the IASA system to monitor collected tweets of pictures in the social media mini cloud during an HA/DR operation, and to get alerted on possible loss of confidentiality, integrity, and availability of the collected social media sensor data. An assumption is made that Detect has already filtered out the status of the monitored target of interest, and the target's feature matrix is produced by Detect software and is available for identify software.

The SoS in the experiment setup is assumed to underutilize the computing hardware resources and the detection module of the DIPR model is deployed within the smart cloud SoS. There will be four test cases as follows:

- Normal operation
- Loss of confidentiality
- Loss of integrity
- Loss of availability

First, in the scenarios where the system is running normally, there is no loss of confidentiality, integrity, and availability of the collected social media sensor data. The second test case involves the loss of confidentiality of the collected social media sensor data where there is an unauthorized access of the data. The third test case involves the loss of integrity of the collected social media sensor data where there is an unauthorized modification of the data. The last test case involves the loss of availability of the collected social media sensor data where there is an unauthorized deletion of the data. Figure 54 provides the schematic diagram of the proof-of-concept experimental setup.

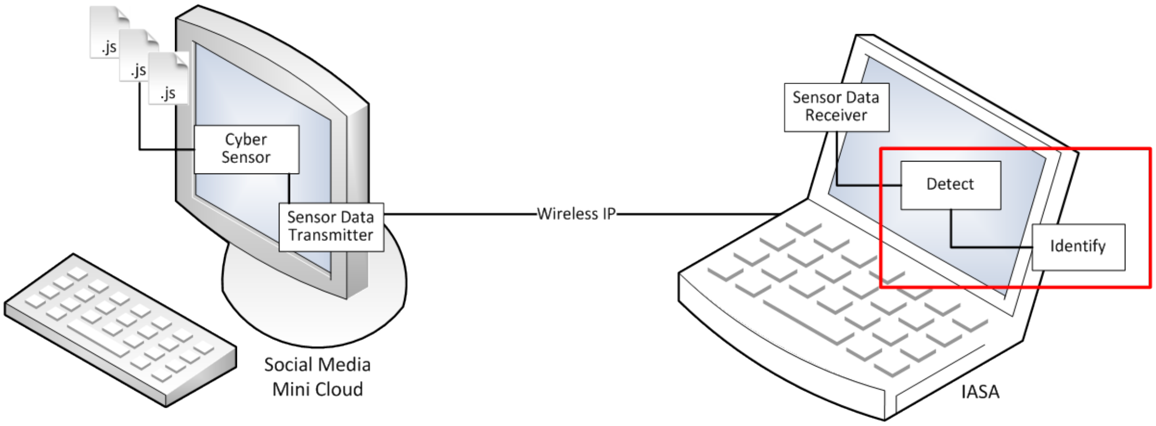


Figure 54. Schematic diagram of the proof-of-concept experimental setup.

The experimental setup involves a computer representing the social media mini cloud and laptop housing the Detect and Identify software of the IASA. Social media files in the form of Twitter JSON files are collected and stored within the mini cloud hard drive; some of the Twitter files contains pictures in the form of JPEG files. The cyber sensor used by IASA will monitor the status of the JPEG files for confidentiality, integrity, and availability. The status of the JPEG files is transmitted using wireless Internet protocols to the laptop. The JPEG documents are received by the sensor data receiver and pushed to the Detect software for analysis. After analysis of the received text documents by the Detect software, the output in the form of the feature matrix or XML files, is pushed to the Identify software for comparison against intelligent state rules. The feature matrix is further updated by the Identify software to reflect any change in state. The proof of concept will cover the portion whereby the Identify software receives the feature matrix from the Detect software and updates it against intelligent state rules, as highlighted in red in Figure 54.

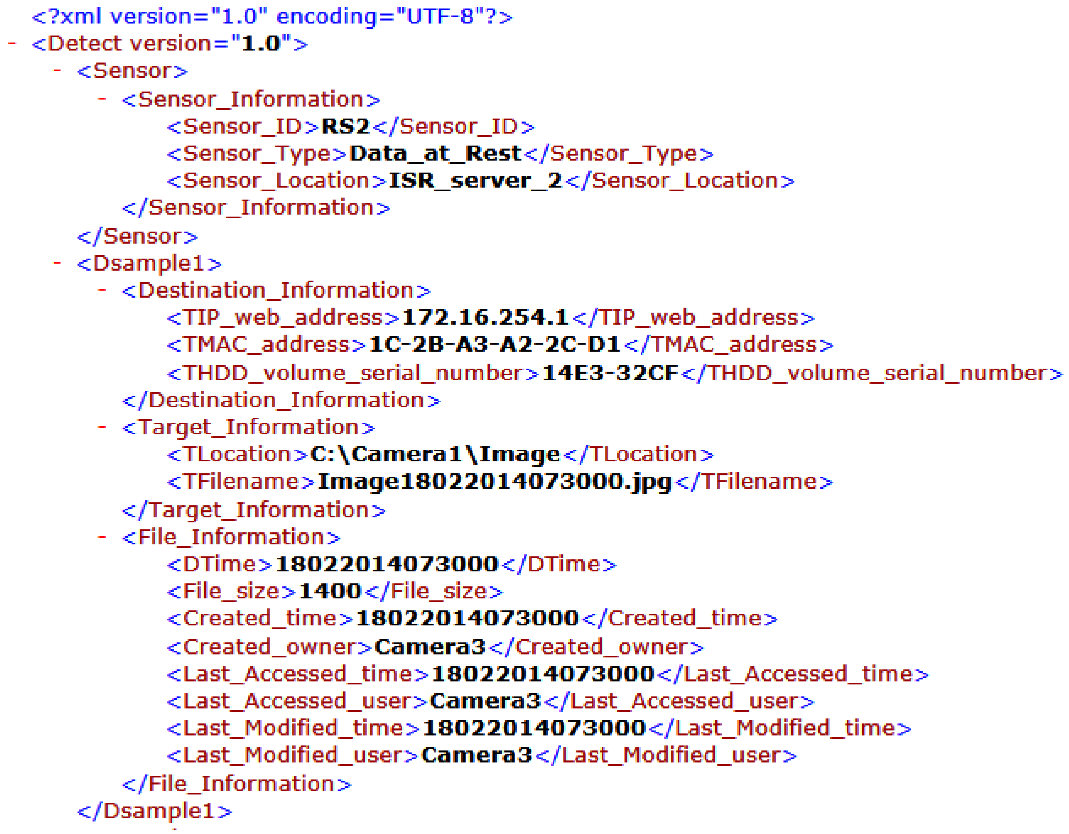
B. SOFTWARE PSEUDO CODE

The Identify software reads the feature matrix XML files produced by the Detect software of the IASA system and outputs the Identify feature matrix XML file. The feature matrix XML file produced by the Detect software is used for the four experimental scenarios discussed in the following section. The Identify MATLAB codes are also discussed.

1. Test Cases Detect Feature Matrix

a. Normal Operation

Figure 55 shows the XML file from the detection software, which is a snapshot showing the information collected from the target. Due to the length of the XML file, only the information from the first detection sample is shown. The full detection XML file can be found in Appendix A.

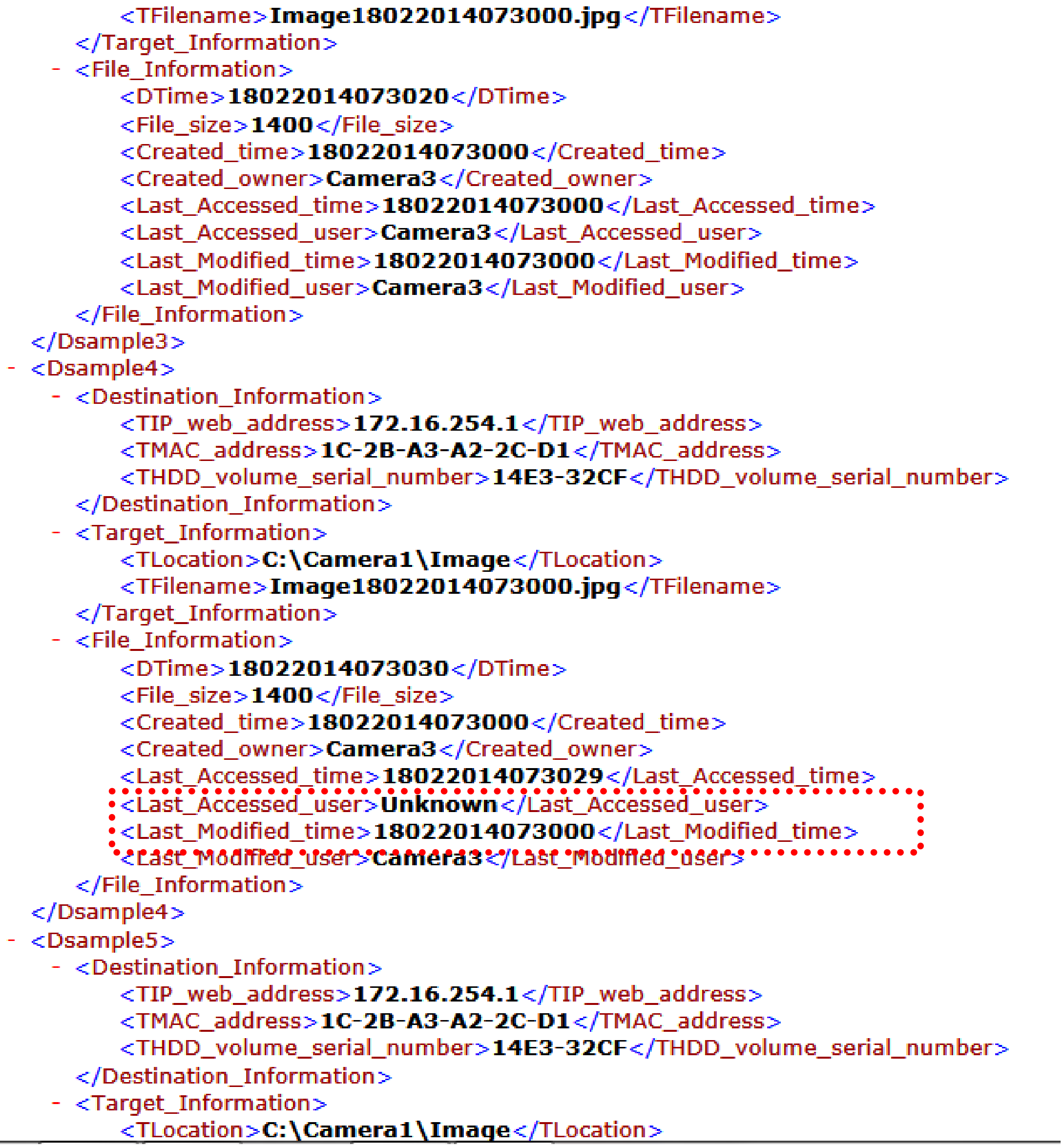


```
  <?xml version="1.0" encoding="UTF-8"?>
- <Detect version="1.0">
  - <Sensor>
    - <Sensor_Information>
        <Sensor_ID>RS2</Sensor_ID>
        <Sensor_Type>Data_at_Rest</Sensor_Type>
        <Sensor_Location>ISR_server_2</Sensor_Location>
      </Sensor_Information>
    </Sensor>
  - <Dsample1>
    - <Destination_Information>
        <TIP_web_address>172.16.254.1</TIP_web_address>
        <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
        <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
        <TLocation>C:\Camera1\Image</TLocation>
        <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
    - <File_Information>
        <DTime>18022014073000</DTime>
        <File_size>1400</File_size>
        <Created_time>18022014073000</Created_time>
        <Created_owner>Camera3</Created_owner>
        <Last_Accessed_time>18022014073000</Last_Accessed_time>
        <Last_Accessed_user>Camera3</Last_Accessed_user>
        <Last_Modified_time>18022014073000</Last_Modified_time>
        <Last_Modified_user>Camera3</Last_Modified_user>
      </File_Information>
    </Dsample1>
```

Figure 55. XML file from the detection software for a normal case.

b. Loss of Confidentiality

The detection feature matrix was modified to reflect the loss of confidentiality in the collected data. Loss of confidentiality results when information is accessed by unauthorized users. The changes (highlighted in Red) in the collected data of the detection feature matrix are shown in Figure 56. The Detect XML file has the fields “last accessed time” and “last accessed user” changed to reflect the case of loss of confidentiality.

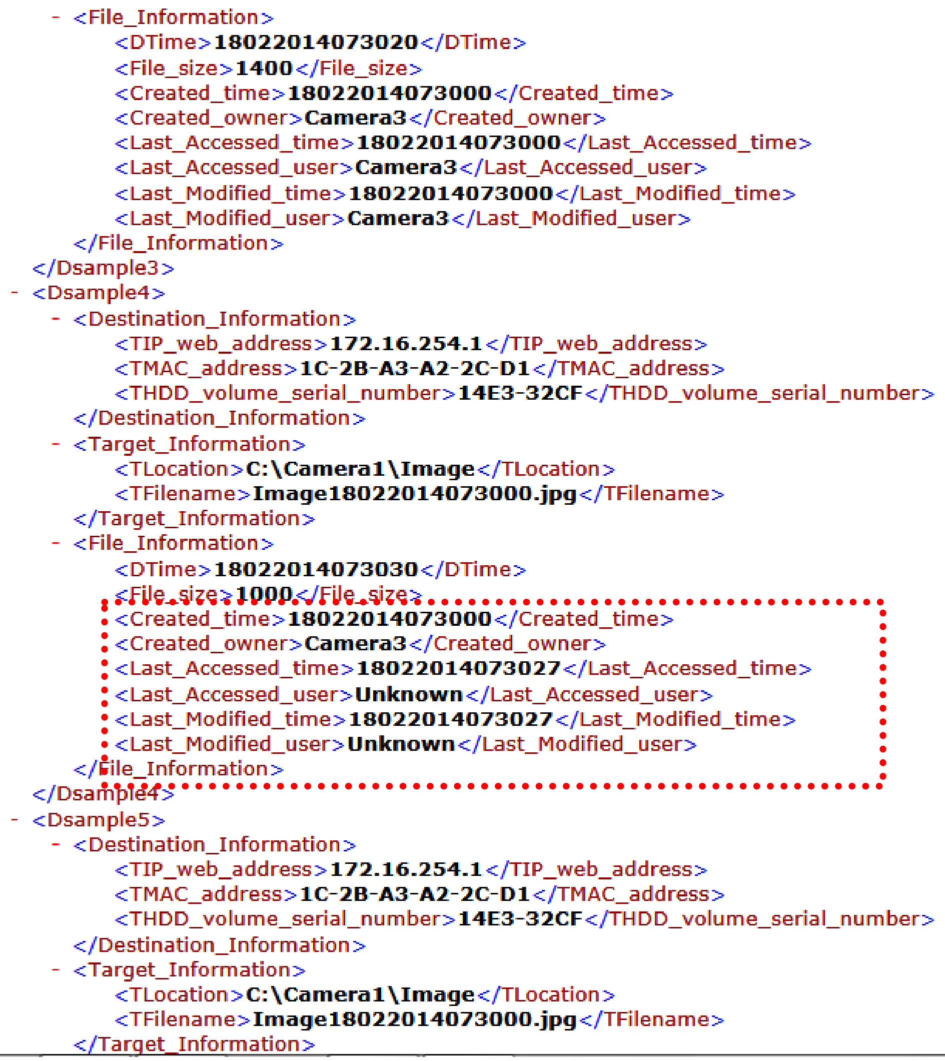


```
        <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
    - <File_Information>
        <DTime>18022014073020</DTime>
        <File_size>1400</File_size>
        <Created_time>18022014073000</Created_time>
        <Created_owner>Camera3</Created_owner>
        <Last_Accessed_time>18022014073000</Last_Accessed_time>
        <Last_Accessed_user>Camera3</Last_Accessed_user>
        <Last_Modified_time>18022014073000</Last_Modified_time>
        <Last_Modified_user>Camera3</Last_Modified_user>
      </File_Information>
  </Dsample3>
- <Dsample4>
    - <Destination_Information>
        <TIP_web_address>172.16.254.1</TIP_web_address>
        <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
        <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
        <TLocation>C:\Camera1\Image</TLocation>
        <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
    - <File_Information>
        <DTime>18022014073030</DTime>
        <File_size>1400</File_size>
        <Created_time>18022014073000</Created_time>
        <Created_owner>Camera3</Created_owner>
        <Last_Accessed_time>18022014073029</Last_Accessed_time>
        <Last_Accessed_user>Unknown</Last_Accessed_user>
        <Last_Modified_time>18022014073000</Last_Modified_time>
        <Last_Modified_user>Camera3</Last_Modified_user>
      </File_Information>
  </Dsample4>
- <Dsample5>
    - <Destination_Information>
        <TIP_web_address>172.16.254.1</TIP_web_address>
        <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
        <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
        <TLocation>C:\Camera1\Image</TLocation>
```

Figure 56. XML file from the detection software for loss of confidentiality.

c. *Loss of Integrity*

The detection feature matrix was modified to reflect the loss of integrity in the collected data. Loss of integrity refers to the trustworthiness of the data and whether it has been tampered with or corrupted by unauthorized users. For a file to be modified it has to be accessed first; therefore, any unauthorized modification to the data is considered both loss of confidentiality and loss of integrity. The changes (highlighted in red) in the collected data of the detection feature matrix are shown in Figure 57. The Detect XML file has the fields “file size,” “last accessed time,” and “last accessed user,” and “last modified time” and “last modified user” changed to reflect the case of loss of integrity.

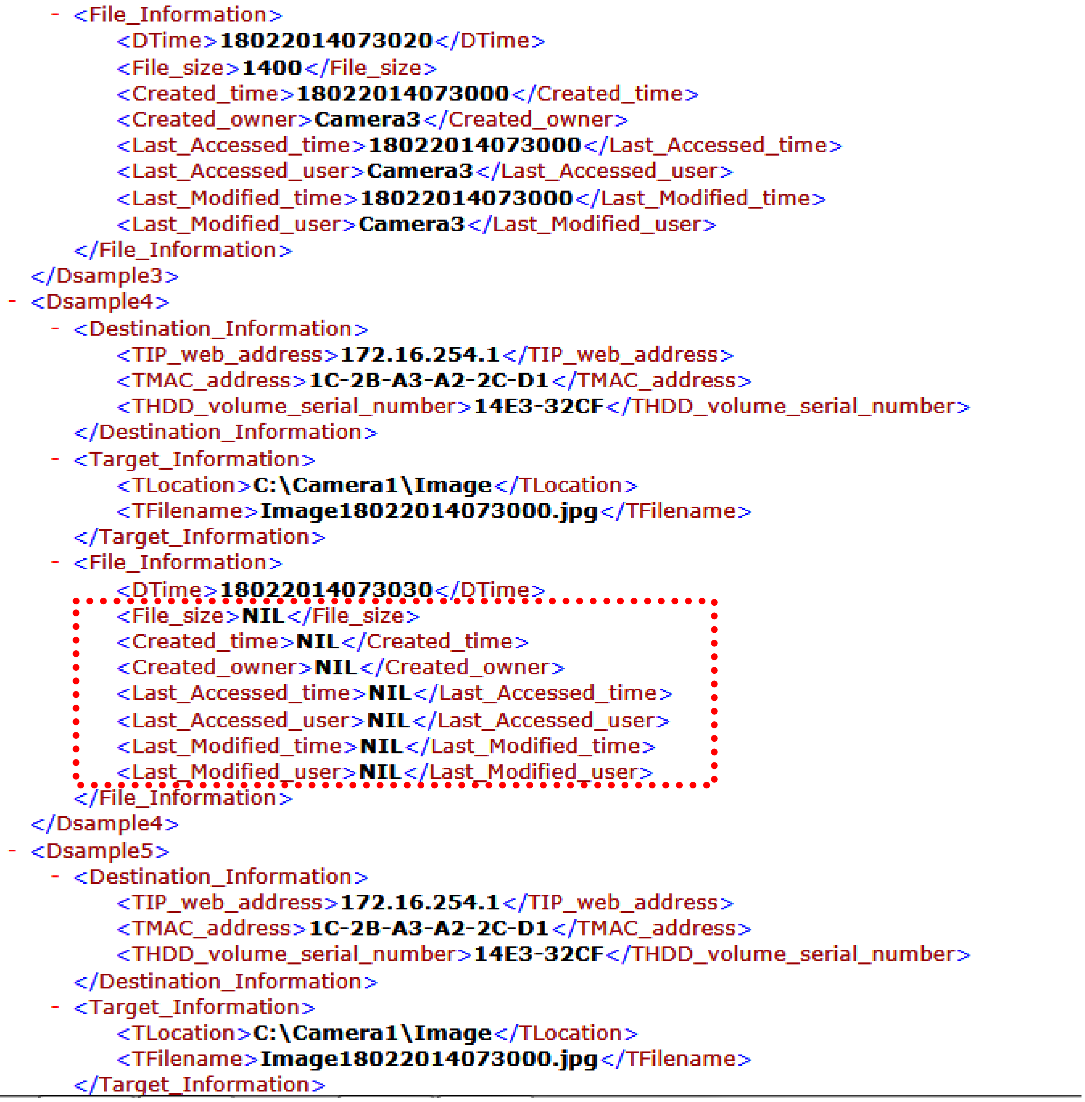


```
    - <File_Information>
          <DTime>18022014073020</DTime>
          <File_size>1400</File_size>
          <Created_time>18022014073000</Created_time>
          <Created_owner>Camera3</Created_owner>
          <Last_Accessed_time>18022014073000</Last_Accessed_time>
          <Last_Accessed_user>Camera3</Last_Accessed_user>
          <Last_Modified_time>18022014073000</Last_Modified_time>
          <Last_Modified_user>Camera3</Last_Modified_user>
      </File_Information>
  </Dsample3>
- <Dsample4>
    - <Destination_Information>
          <TIP_web_address>172.16.254.1</TIP_web_address>
          <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
          <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
          <TLocation>C:\Camera1\Image</TLocation>
          <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
    - <File_Information>
          <DTime>18022014073030</DTime>
          <File_size>1000</File_size>
          <Created_time>18022014073000</Created_time>
          <Created_owner>Camera3</Created_owner>
          <Last_Accessed_time>18022014073027</Last_Accessed_time>
          <Last_Accessed_user>Unknown</Last_Accessed_user>
          <Last_Modified_time>18022014073027</Last_Modified_time>
          <Last_Modified_user>Unknown</Last_Modified_user>
      </File_Information>
  </Dsample4>
- <Dsample5>
    - <Destination_Information>
          <TIP_web_address>172.16.254.1</TIP_web_address>
          <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
          <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
          <TLocation>C:\Camera1\Image</TLocation>
          <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
```

Figure 57. XML file from the detection software for loss of integrity.

d. Loss of Availability

The detection feature matrix was modified to reflect the loss of availability in the collected data. Loss of availability refers to the inability of the user to access the data when required; the file has been renamed or removed from the monitored location. The changes (highlighted in red) in the collected data of the detection feature matrix are shown in Figure 58. The Detect XML file has the all the fields reported NIL to reflect the case of a missing file.



```
    - <File_Information>
        <DTime>18022014073020</DTime>
        <File_size>1400</File_size>
        <Created_time>18022014073000</Created_time>
        <Created_owner>Camera3</Created_owner>
        <Last_Accessed_time>18022014073000</Last_Accessed_time>
        <Last_Accessed_user>Camera3</Last_Accessed_user>
        <Last_Modified_time>18022014073000</Last_Modified_time>
        <Last_Modified_user>Camera3</Last_Modified_user>
      </File_Information>
  </Dsample3>
- <Dsample4>
    - <Destination_Information>
        <TIP_web_address>172.16.254.1</TIP_web_address>
        <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
        <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
        <TLocation>C:\Camera1\Image</TLocation>
        <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
    - <File_Information>
        <DTime>18022014073030</DTime>
        <File_size>NIL</File_size>
        <Created_time>NIL</Created_time>
        <Created_owner>NIL</Created_owner>
        <Last_Accessed_time>NIL</Last_Accessed_time>
        <Last_Accessed_user>NIL</Last_Accessed_user>
        <Last_Modified_time>NIL</Last_Modified_time>
        <Last_Modified_user>NIL</Last_Modified_user>
      </File_Information>
  </Dsample4>
- <Dsample5>
    - <Destination_Information>
        <TIP_web_address>172.16.254.1</TIP_web_address>
        <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
        <THDD_volume_serial_number>14E3-32CF</THDD_volume_serial_number>
      </Destination_Information>
    - <Target_Information>
        <TLocation>C:\Camera1\Image</TLocation>
        <TFilename>Image18022014073000.jpg</TFilename>
      </Target_Information>
```

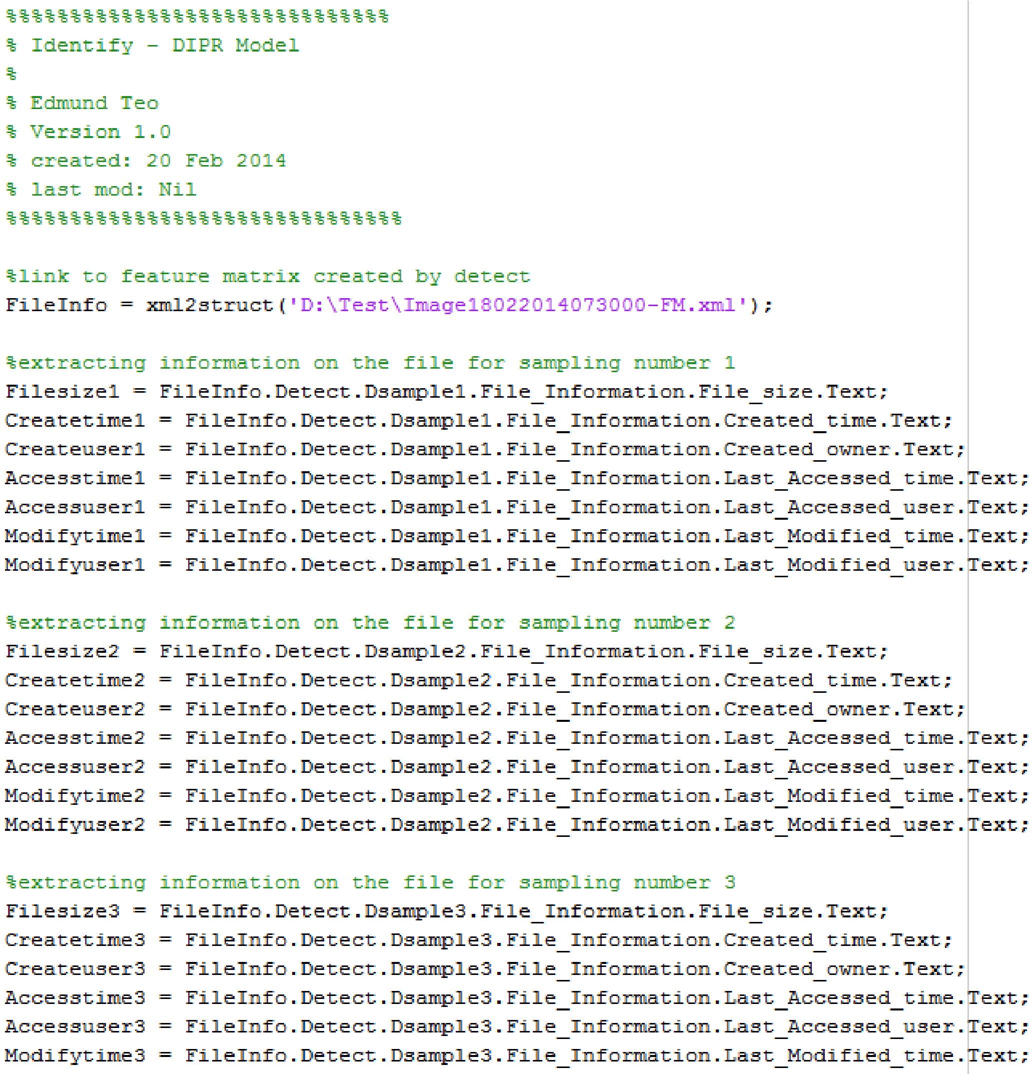
Figure 58. XML file from the detection software for loss of availability.

2. Identify MATLAB Coding

The Identify software is made up of two MATLAB codes. The first is the component to sieve and filter out the status of the target file using the information collected from the Detect software, in the form of the detect feature matrix XML file. The second portion of the Identify software writes the identify status of the monitored target file into the identify feature matrix XML file. These two MATLAB codes are discussed in the following section.

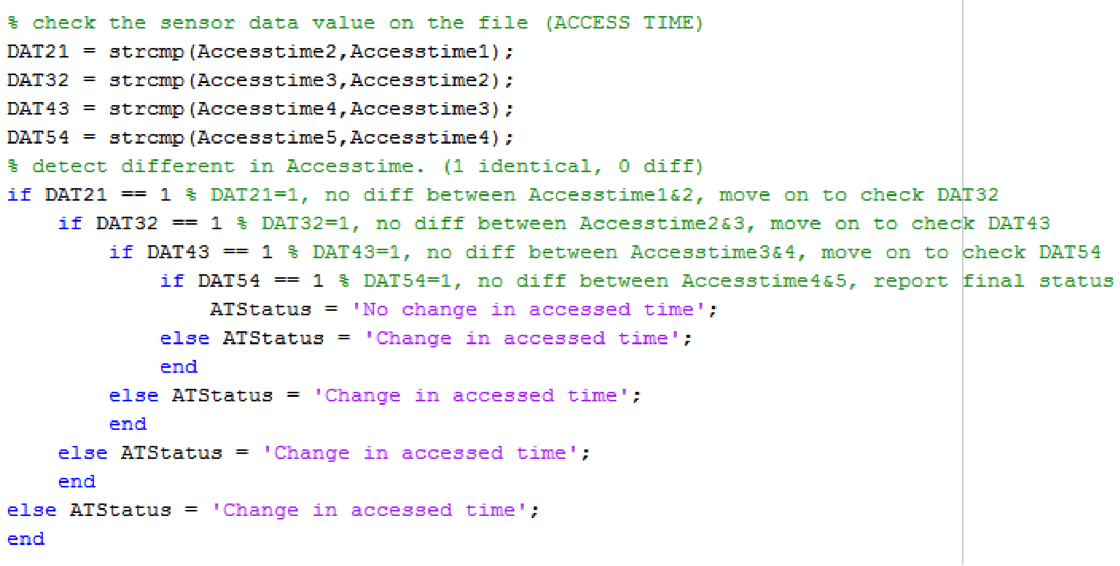
a. Identify

The Identify code makes use of Wouter Falkena's xml2struct function to read and extract the data from the detect feature matrix XML file. The information is then assigned individually as a variable in MATLAB (Figure 59). After assigning the variable in MATLAB, the data are compared for any discrepancy between the sampled detection time, such as changes in file size or timestamp and user access. The timestamp function from Eddie is used to create the identify time variable in MATLAB; this will be subsequently used to reflect the time identification software was run in the identify feature matrix XML file. A snapshot of the code is shown in Figure 60 (Eddie 2012). The full Identify MATLAB code can be found in Appendix B.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identify - DIPR Model
%
% Edmund Teo
% Version 1.0
% created: 20 Feb 2014
% last mod: Nil
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%link to feature matrix created by detect
FileInfo = xml2struct('D:\Test\Image18022014073000-FM.xml');

%extracting information on the file for sampling number 1
Filesize1 = FileInfo.Detect.Dsample1.File_Information.File_size.Text;
Createtime1 = FileInfo.Detect.Dsample1.File_Information.Created_time.Text;
Createuser1 = FileInfo.Detect.Dsample1.File_Information.Created_owner.Text;
Accesstime1 = FileInfo.Detect.Dsample1.File_Information.Last_Accessed_time.Text;
Accessuser1 = FileInfo.Detect.Dsample1.File_Information.Last_Accessed_user.Text;
Modifytime1 = FileInfo.Detect.Dsample1.File_Information.Last_Modified_time.Text;
Modifyuser1 = FileInfo.Detect.Dsample1.File_Information.Last_Modified_user.Text;

%extracting information on the file for sampling number 2
Filesize2 = FileInfo.Detect.Dsample2.File_Information.File_size.Text;
Createtime2 = FileInfo.Detect.Dsample2.File_Information.Created_time.Text;
Createuser2 = FileInfo.Detect.Dsample2.File_Information.Created_owner.Text;
Accesstime2 = FileInfo.Detect.Dsample2.File_Information.Last_Accessed_time.Text;
Accessuser2 = FileInfo.Detect.Dsample2.File_Information.Last_Accessed_user.Text;
Modifytime2 = FileInfo.Detect.Dsample2.File_Information.Last_Modified_time.Text;
Modifyuser2 = FileInfo.Detect.Dsample2.File_Information.Last_Modified_user.Text;

%extracting information on the file for sampling number 3
Filesize3 = FileInfo.Detect.Dsample3.File_Information.File_size.Text;
Createtime3 = FileInfo.Detect.Dsample3.File_Information.Created_time.Text;
Createuser3 = FileInfo.Detect.Dsample3.File_Information.Created_owner.Text;
Accesstime3 = FileInfo.Detect.Dsample3.File_Information.Last_Accessed_time.Text;
Accessuser3 = FileInfo.Detect.Dsample3.File_Information.Last_Accessed_user.Text;
Modifytime3 = FileInfo.Detect.Dsample3.File_Information.Last_Modified_time.Text;
```

Figure 59. MATLAB coding for Identify, extracting data from XML and assigning variables to the data.

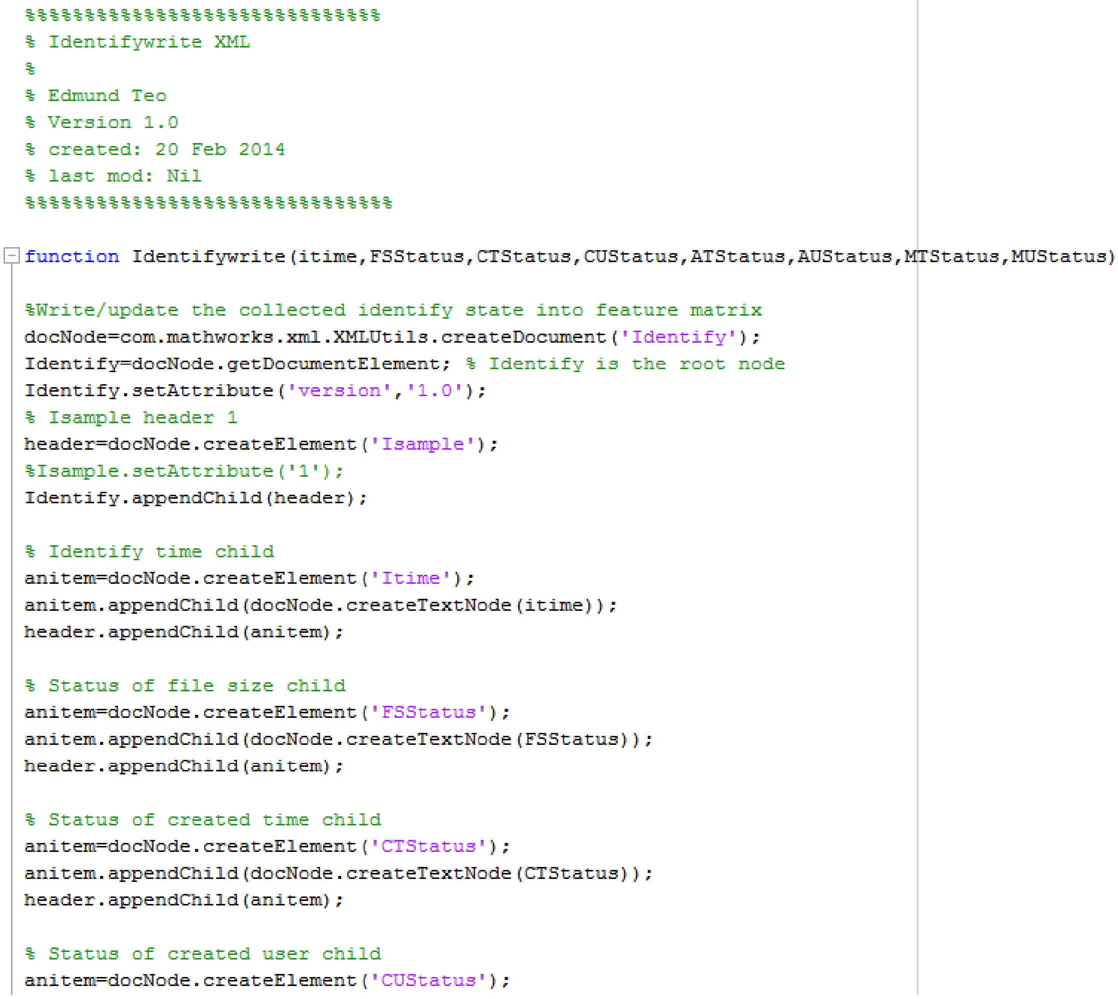


```
% check the sensor data value on the file (ACCESS TIME)
DAT21 = strcmp(Accesstime2,Accesstime1);
DAT32 = strcmp(Accesstime3,Accesstime2);
DAT43 = strcmp(Accesstime4,Accesstime3);
DAT54 = strcmp(Accesstime5,Accesstime4);
% detect different in Accesstime. (1 identical, 0 diff)
if DAT21 == 1 % DAT21=1, no diff between Accesstime1&2, move on to check DAT32
    if DAT32 == 1 % DAT32=1, no diff between Accesstime2&3, move on to check DAT43
        if DAT43 == 1 % DAT43=1, no diff between Accesstime3&4, move on to check DAT54
            if DAT54 == 1 % DAT54=1, no diff between Accesstime4&5, report final status
                ATStatus = 'No change in accessed time';
            else ATStatus = 'Change in accessed time';
            end
        else ATStatus = 'Change in accessed time';
        end
    else ATStatus = 'Change in accessed time';
    end
else ATStatus = 'Change in accessed time';
end
```

Figure 60. MATLAB coding for Identify, comparing the data extracted from detect XML.

b. Identify XML Write

The Identify XML Write code makes use of the MATLAB XML write function to create the identify feature matrix XML file. The status of the target file is input as a sub-tag within the Identify XML. A snapshot of the code is shown in Figure 61, and the complete MATLAB coding can be found in Appendix C.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identifywrite XML
%
% Edmund Teo
% Version 1.0
% created: 20 Feb 2014
% last mod: Nil
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Identifywrite(itime,FSStatus,CTStatus,CUStatus,ATStatus,AUStatus,MTStatus,MUStatus)

%Write/update the collected identify state into feature matrix
docNode=com.mathworks.xml.XMLUtils.createDocument('Identify');
Identify=docNode.getDocumentElement; % Identify is the root node
Identify.setAttribute('version','1.0');
% Isample header 1
header=docNode.createElement('Isample');
%Isample.setAttribute('1');
Identify.appendChild(header);

% Identify time child
anitem=docNode.createElement('Itime');
anitem.appendChild(docNode.createTextNode(itime));
header.appendChild(anitem);

% Status of file size child
anitem=docNode.createElement('FSStatus');
anitem.appendChild(docNode.createTextNode(FSStatus));
header.appendChild(anitem);

% Status of created time child
anitem=docNode.createElement('CTStatus');
anitem.appendChild(docNode.createTextNode(CTStatus));
header.appendChild(anitem);

% Status of created user child
anitem=docNode.createElement('CUStatus');
```

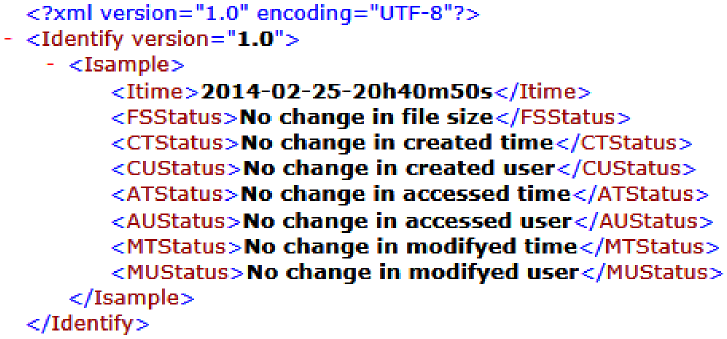
Figure 61. MATLAB coding for Identify XML Write.

C. RESULTS

The results from the Identify module for the four test cases are discussed in this section.

1. Normal Operation

In a normal monitoring situation, the target file after creation should not be accessed or modified by any user other than the file creator. If it has, this can be determined by examining the timestamp of the monitored file, along with the created, accessed, and modified time. In addition, there should not be another user who has accessed or modified the file besides the creator. The file size of the target should be consistently the same size throughout the monitored period. The identify feature matrix under a normal monitoring situation is shown in Figure 62. The identify feature matrix provides the status of the target file by examining the detection sampling results and comparing the information for any changes in file size, timestamp, and user.

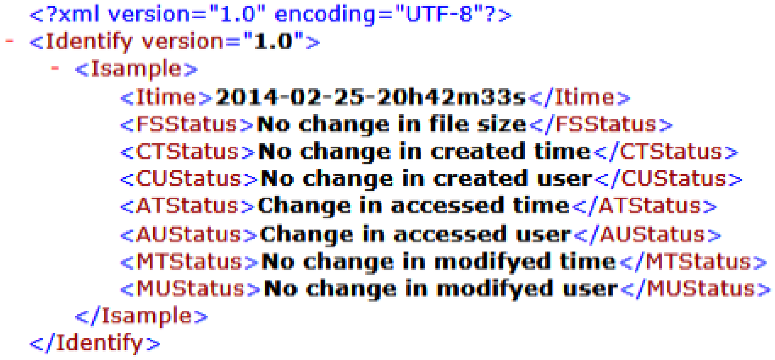


```
  <?xml version="1.0" encoding="UTF-8"?>
- <Identify version="1.0">
  - <Isample>
      <Itime>2014-02-25-20h40m50s</Itime>
      <FSStatus>No change in file size</FSStatus>
      <CTStatus>No change in created time</CTStatus>
      <CUStatus>No change in created user</CUStatus>
      <ATStatus>No change in accessed time</ATStatus>
      <AUStatus>No change in accessed user</AUStatus>
      <MTStatus>No change in modifyed time</MTStatus>
      <MUStatus>No change in modifyed user</MUStatus>
    </Isample>
  </Identify>
```

Figure 62. XML output from Identify module for normal case.

2. Loss of Confidentiality

The confidentiality of a file can be assessed by examining the status of the file’s last accessed time and last accessed user. A difference between the detection samples is an indication of unauthorized access to the monitored file. Figure 63 shows the output of the Identify software identifying the difference in the last accessed time and user of the file.

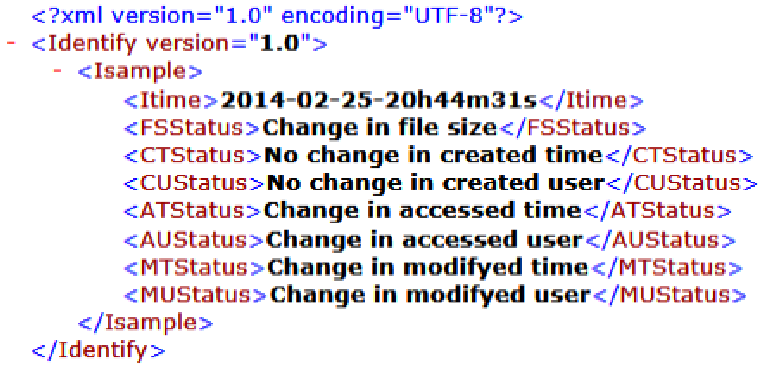


```
  <?xml version="1.0" encoding="UTF-8"?>
- <Identify version="1.0">
  - <Isample>
      <Itime>2014-02-25-20h42m33s</Itime>
      <FSStatus>No change in file size</FSStatus>
      <CTStatus>No change in created time</CTStatus>
      <CUStatus>No change in created user</CUStatus>
      <ATStatus>Change in accessed time</ATStatus>
      <AUStatus>Change in accessed user</AUStatus>
      <MTStatus>No change in modifyed time</MTStatus>
      <MUStatus>No change in modifyed user</MUStatus>
    </Isample>
  </Identify>
```

Figure 63. XML output from Identify module for loss of confidentiality case.

3. Loss of Integrity

The integrity of a file can be assessed by examining the status of the file's last modified time and last modified user. A difference between the detection samples is an indication of an unauthorized modification to the monitored file. As previously explained, a file has to be accessed in order to be modified; therefore, the last accessed time and user will also be changed. Figure 64 shows the output of the Identify software identifying the difference in the file size, last accessed time and user, and last modified time and user of the file.

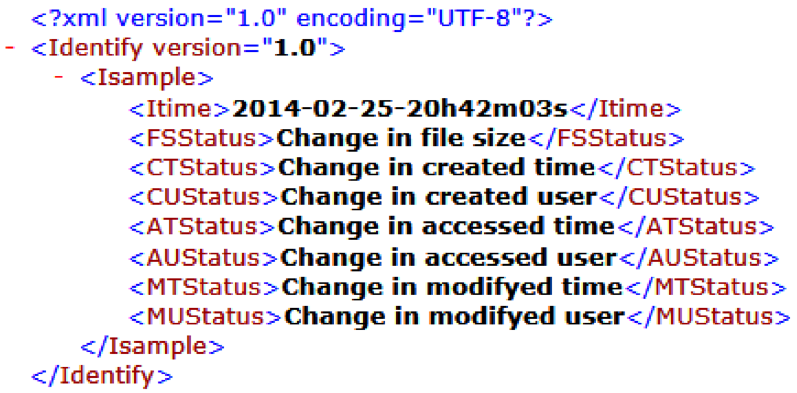


```
  <?xml version="1.0" encoding="UTF-8"?>
- <Identify version="1.0">
  - <Isample>
        <Itime>2014-02-25-20h44m31s</Itime>
        <FSStatus>Change in file size</FSStatus>
        <CTStatus>No change in created time</CTStatus>
        <CUStatus>No change in created user</CUStatus>
        <ATStatus>Change in accessed time</ATStatus>
        <AUStatus>Change in accessed user</AUStatus>
        <MTStatus>Change in modifyed time</MTStatus>
        <MUStatus>Change in modifyed user</MUStatus>
    </Isample>
  </Identify>
```

Figure 64. XML output from Identify module for loss of integrity case.

4. Loss of Availability

The availability of a file can be assessed by examining any status of the file. A missing file will not be detected by the Detect software; therefore, the missing status is recorded in the detect feature matrix XML file. Figure 65 shows the output of the Identify software identifying the difference in all fields as the file was no longer detected, therefore, recording a NIL return.



```
  <?xml version="1.0" encoding="UTF-8"?>
- <Identify version="1.0">
  - <Isample>
      <Itime>2014-02-25-20h42m03s</Itime>
      <FSStatus>Change in file size</FSStatus>
      <CTStatus>Change in created time</CTStatus>
      <CUStatus>Change in created user</CUStatus>
      <ATStatus>Change in accessed time</ATStatus>
      <AUStatus>Change in accessed user</AUStatus>
      <MTStatus>Change in modifyed time</MTStatus>
      <MUStatus>Change in modifyed user</MUStatus>
    </Isample>
  </Identify>
```

Figure 65. XML output from Identify module for loss of availability case.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS, RECOMMENDATIONS, AND AREAS FOR FURTHER RESEARCH

Possible improvements for the proof of concept, suggestions for future work which would create a more robust system, and final conclusions on the success of the overall system are covered in this chapter.

A. CONCLUSIONS

1. IASA System

The proposed IASA system proves to be able to ensure automated information assurance for cyber situational awareness of information with the assurance of data confidentially, integrity and availability through the use of an intelligence automation system. The successful development and deployment, with the CSAC2 and CADSA system, in the cloud environment will provide real-time assessment of any cyber system supporting military operations, automating the handling of big data and filtering the necessary alerts for the operators. Thereby it ensures essential data that are collected are not tampered with, resulting in the loss of information confidentiality, integrity, and availability.

2. Proof of Concept of Identify DIPR Model

The proof of concept successfully implemented the identify DIPR model for the IASA system. Using MATLAB software to read the detect feature matrix XML file, the code was able to extract the required information of the data at rest file, and output the status of the target to an identify feature matrix XML file. The identify coding was put through four scenarios whereby the file confidentiality, integrity, and availability was affected, and the code was able to successfully pick up the changes in status.

B. RECOMMENDATIONS TO IMPROVE THE PROOF OF CONCEPT

After the proof of concept was completed and tested, the author uncovered areas where improvement can be made, which will result in a superior product. While the current proof of concept meets the intended objectives to demonstrate the abilities of the identify DIPR model, there are specific areas that will make the process better.

1. Completeness

If the proof of concept was able to include the detection, prediction, and reaction phase of the DIPR model, it would provide a better view of the IASA system and completeness from the detection of file source to the reaction due to changes in the predetermined intelligent state rules. The proof of concept should as far as possible emulate the actual deployment of the IASA system in the enterprise network. The Detect software should be collecting data on the mini cloud host and transmitting the feature matrixes to the IASA system for analysis over a LAN or wireless network; this setup would expose or highlight any possible issues when transmitting over a live network.

2. Refining the Software Coding

The identify feature matrix used in the proof-of-concept experimental setup wrote its identify result as a new XML file. Ideally, it would be better for a single feature matrix to be constantly updated by the DIPR software. This feature matrix will always be tagged to the monitored target, therefore, allowing better traceability and resources management. This can be done with more complex MATLAB coding to the identify XML write module of the identify software.

C. AREAS FOR FURTHER RESEARCH

Future work to develop and expand the architecture of the proposed system is discussed in this section. The usage of other file characteristics such as monitoring candidates, evaluating the effectiveness of monitoring other state of data, and refining the DIPR model is discussed.

1. Other File Characteristics as Monitoring Candidates

In the proof-of-concept test, only the data state of data at rest has been examined, in which the file characteristics such as timestamps, user, and file size are used as monitoring candidates. Other file characteristics, such as in-depth analysis of the file source codes or binary data, can be employed but this will require complex programming in the Identify software and takes up more computing resources.

The monitoring candidates can be further expanded with the inclusion of other data states, such as data in use and data in transit; each state requires a unique methodology of analyzing and monitoring the data, therefore ensuring information assurance is upheld in the system.

2. Effectiveness of Monitoring Data in Transit and in Use

Monitoring of data in transit poses a different set of monitoring candidates, sensors, and deployment method. An additional set of computing resources has to be setup to analyze data movement across the network. One method is the use of packet analyzer software to determine the data transmitted retains its confidentiality, integrity, and availability.

Data in use is the most difficult state of data to monitor; the data can be constantly in use by users, and sensors have to be able to determine which changes are authorized and which are not. Deployment of numerous sensors with high sampling rates can bog down the host's computing system, eating up valuable computing resources needed. Development of sensors for data in use can be further analyzed and studied.

3. Expanding and Integrating with the Proposed CSAC2 and CADSA System

Additional research and experimental testing is required to validate the trio combination setup and concept of CSAC2, CADSA, and IASA system within an enterprise network. The interfaces and interactions between these three systems need to be further standardized to ensure commands and alerts are understood between these systems; a common language, such as the usage of XML, can be explored to achieve this.

The complexity of these systems will surely create opportunities for further research into areas such as software development, communication protocol, standardization of file structure, self-protection against external and internal cyber attack, and human system interfaces. The effectiveness of the synergy and information sharing among the trio systems can also be further explored.

APPENDIX A. DETECT FEATURE MATRIX

```
<?xml version="1.0" encoding="utf-8"?>
<Detect version="1.0">
      <Sensor>
            <Sensor_Information>
                  <Sensor_ID>RS2</Sensor_ID>
                  <Sensor_Type>Data_at_Rest</Sensor_Type>
                  <Sensor_Location>ISR_server_2</Sensor_Location>
            </Sensor_Information>
      </Sensor>
      <Dsample1>
            <Destination_Information>
                  <TIP_web_address>172.16.254.1</TIP_web_address>
                  <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
                  <THDD_volume_serial_number>14E3-
32CF</THDD_volume_serial_number>
            </Destination_Information>
            <Target_Information>
                  <TLocation>C:\Camera1\Image</TLocation>
                  <TFilename>Image18022014073000.jpg</TFilename>
            </Target_Information>
            <File_Information>
                  <DTime>18022014073000</DTime>
                  <File_size>1400</File_size>
                  <Created_time>18022014073000</Created_time>
                  <Created_owner>Camera3</Created_owner>

      <Last_Accessed_time>18022014073000</Last_Accessed_time>
                  <Last_Accessed_user>Camera3</Last_Accessed_user>

      <Last_Modified_time>18022014073000</Last_Modified_time>
                  <Last_Modified_user>Camera3</Last_Modified_user>
            </File_Information>
      </Dsample1>
      <Dsample2>
            <Destination_Information>
                  <TIP_web_address>172.16.254.1</TIP_web_address>
                  <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
                  <THDD_volume_serial_number>14E3-
32CF</THDD_volume_serial_number>
            </Destination_Information>
            <Target_Information>
                  <TLocation>C:\Camera1\Image</TLocation>
                  <TFilename>Image18022014073000.jpg</TFilename>
            </Target_Information>
            <File_Information>
                  <DTime>18022014073010</DTime>
                  <File_size>1400</File_size>
                  <Created_time>18022014073000</Created_time>
                  <Created_owner>Camera3</Created_owner>

      <Last_Accessed_time>18022014073000</Last_Accessed_time>
```

```
                  <Last_Accessed_user>Camera3</Last_Accessed_user>

      <Last_Modified_time>18022014073000</Last_Modified_time>
                  <Last_Modified_user>Camera3</Last_Modified_user>
            </File_Information>
      </Dsample2>
      <Dsample3>
            <Destination_Information>
                  <TIP_web_address>172.16.254.1</TIP_web_address>
                  <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
                  <THDD_volume_serial_number>14E3-
32CF</THDD_volume_serial_number>
            </Destination_Information>
            <Target_Information>
                  <TLocation>C:\Camera1\Image</TLocation>
                  <TFilename>Image18022014073000.jpg</TFilename>
            </Target_Information>
            <File_Information>
                  <DTime>18022014073020</DTime>
                  <File_size>1400</File_size>
                  <Created_time>18022014073000</Created_time>
                  <Created_owner>Camera3</Created_owner>

      <Last_Accessed_time>18022014073000</Last_Accessed_time>
                  <Last_Accessed_user>Camera3</Last_Accessed_user>

      <Last_Modified_time>18022014073000</Last_Modified_time>
                  <Last_Modified_user>Camera3</Last_Modified_user>
            </File_Information>
      </Dsample3>
      <Dsample4>
            <Destination_Information>
                  <TIP_web_address>172.16.254.1</TIP_web_address>
                  <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
                  <THDD_volume_serial_number>14E3-
32CF</THDD_volume_serial_number>
            </Destination_Information>
            <Target_Information>
                  <TLocation>C:\Camera1\Image</TLocation>
                  <TFilename>Image18022014073000.jpg</TFilename>
            </Target_Information>
            <File_Information>
                  <DTime>18022014073030</DTime>
                  <File_size>1400</File_size>
                  <Created_time>18022014073000</Created_time>
                  <Created_owner>Camera3</Created_owner>

      <Last_Accessed_time>18022014073000</Last_Accessed_time>
                  <Last_Accessed_user>Camera3</Last_Accessed_user>

      <Last_Modified_time>18022014073000</Last_Modified_time>
                  <Last_Modified_user>Camera3</Last_Modified_user>
            </File_Information>
      </Dsample4>
      <Dsample5>
```

```
            <Destination_Information>
                  <TIP_web_address>172.16.254.1</TIP_web_address>
                  <TMAC_address>1C-2B-A3-A2-2C-D1</TMAC_address>
                  <THDD_volume_serial_number>14E3-
32CF</THDD_volume_serial_number>
            </Destination_Information>
            <Target_Information>
                  <TLocation>C:\Camera1\Image</TLocation>
                  <TFilename>Image18022014073000.jpg</TFilename>
            </Target_Information>
            <File_Information>
                  <DTime>18022014073040</DTime>
                  <File_size>1400</File_size>
                  <Created_time>18022014073000</Created_time>
                  <Created_owner>Camera3</Created_owner>

      <Last_Accessed_time>18022014073000</Last_Accessed_time>
                  <Last_Accessed_user>Camera3</Last_Accessed_user>

      <Last_Modified_time>18022014073000</Last_Modified_time>
                  <Last_Modified_user>Camera3</Last_Modified_user>
            </File_Information>
      </Dsample5>
</Detect>
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. IDENTIFY DIPR MODEL MATLAB CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identify - DIPR Model
%
% Edmund Teo
% Version 1.0
% created: 20 Feb 2014
% last mod: Nil
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%link to feature matrix created by detect
FileInfo = xml2struct('D:\Test\Image18022014073000-FM.xml');

%extracting information on the file for sampling number 1
Filesize1 = FileInfo.Detect.Dsample1.File_Information.File_size.Text;
Createtime1 =
FileInfo.Detect.Dsample1.File_Information.Created_time.Text;
Createuser1 =
FileInfo.Detect.Dsample1.File_Information.Created_owner.Text;
Accesstime1 =
FileInfo.Detect.Dsample1.File_Information.Last_Accessed_time.Text;
Accessuser1 =
FileInfo.Detect.Dsample1.File_Information.Last_Accessed_user.Text;
Modifytime1 =
FileInfo.Detect.Dsample1.File_Information.Last_Modified_time.Text;
Modifyuser1 =
FileInfo.Detect.Dsample1.File_Information.Last_Modified_user.Text;

%extracting information on the file for sampling number 2
Filesize2 = FileInfo.Detect.Dsample2.File_Information.File_size.Text;
Createtime2 =
FileInfo.Detect.Dsample2.File_Information.Created_time.Text;
Createuser2 =
FileInfo.Detect.Dsample2.File_Information.Created_owner.Text;
Accesstime2 =
FileInfo.Detect.Dsample2.File_Information.Last_Accessed_time.Text;
Accessuser2 =
FileInfo.Detect.Dsample2.File_Information.Last_Accessed_user.Text;
Modifytime2 =
FileInfo.Detect.Dsample2.File_Information.Last_Modified_time.Text;
Modifyuser2 =
FileInfo.Detect.Dsample2.File_Information.Last_Modified_user.Text;

%extracting information on the file for sampling number 3
Filesize3 = FileInfo.Detect.Dsample3.File_Information.File_size.Text;
Createtime3 =
FileInfo.Detect.Dsample3.File_Information.Created_time.Text;
Createuser3 =
FileInfo.Detect.Dsample3.File_Information.Created_owner.Text;
Accesstime3 =
FileInfo.Detect.Dsample3.File_Information.Last_Accessed_time.Text;
```

```
Accessuser3 =
FileInfo.Detect.Dsample3.File_Information.Last_Accessed_user.Text;
Modifytime3 =
FileInfo.Detect.Dsample3.File_Information.Last_Modified_time.Text;
Modifyuser3 =
FileInfo.Detect.Dsample3.File_Information.Last_Modified_user.Text;

%extracting information on the file for sampling number 4
Filesize4 = FileInfo.Detect.Dsample4.File_Information.File_size.Text;
Createtime4 =
FileInfo.Detect.Dsample4.File_Information.Created_time.Text;
Createuser4 =
FileInfo.Detect.Dsample4.File_Information.Created_owner.Text;
Accesstime4 =
FileInfo.Detect.Dsample4.File_Information.Last_Accessed_time.Text;
Accessuser4 =
FileInfo.Detect.Dsample4.File_Information.Last_Accessed_user.Text;
Modifytime4 =
FileInfo.Detect.Dsample4.File_Information.Last_Modified_time.Text;
Modifyuser4 =
FileInfo.Detect.Dsample4.File_Information.Last_Modified_user.Text;

%extracting information on the file for sampling number 5
Filesize5 = FileInfo.Detect.Dsample5.File_Information.File_size.Text;
Createtime5 =
FileInfo.Detect.Dsample5.File_Information.Created_time.Text;
Createuser5 =
FileInfo.Detect.Dsample5.File_Information.Created_owner.Text;
Accesstime5 =
FileInfo.Detect.Dsample5.File_Information.Last_Accessed_time.Text;
Accessuser5 =
FileInfo.Detect.Dsample5.File_Information.Last_Accessed_user.Text;
Modifytime5 =
FileInfo.Detect.Dsample5.File_Information.Last_Modified_time.Text;
Modifyuser5 =
FileInfo.Detect.Dsample5.File_Information.Last_Modified_user.Text;

% check the sensor data value on the file (FILE SIZE)
DFS21 = strcmp(Filesize2,Filesize1);
DFS32 = strcmp(Filesize3,Filesize2);
DFS43 = strcmp(Filesize4,Filesize3);
DFS54 = strcmp(Filesize5,Filesize4);
% detect different in Createtime. (1 identical, 0 diff)
if DFS21 == 1 % DFS21=1, no diff between filesize1&2, move on to check
DFS32
    if DFS32 == 1 % DFS32=1, no diff between filesize2&3, move on to
check DFS43
        if DFS43 == 1 % DFS43=1, no diff between filesize3&4, move on
to check DFS54
            if DFS54 == 1 % DFS54=1, no diff between filesize4&5,
report final status
                FSStatus = 'No change in file size';
            else FSStatus = 'Change in file size';
            end
        else FSStatus = 'Change in file size';
        end
    else FSStatus = 'Change in file size';
    end
else FSStatus = 'Change in file size';
end

% check the sensor data value on the file (CREATE TIME)
DCT21 = strcmp(Createtime2,Createtime1);
DCT32 = strcmp(Createtime3,Createtime2);
DCT43 = strcmp(Createtime4,Createtime3);
DCT54 = strcmp(Createtime5,Createtime4);
% detect different in Createtime. (1 identical, 0 diff)
if DCT21 == 1 % DCT21=1, no diff between createtime1&2, move on to
check DCT32
    if DCT32 == 1 % DCT32=1, no diff between createtime2&3, move on to
check DCT43
        if DCT43 == 1 % DCT43=1, no diff between createtime3&4, move on
to check DCT54
            if DCT54 == 1 % DCT54=1, no diff between createtime4&5,
report final status
                CTStatus = 'No change in created time';
            else CTStatus = 'Change in created time';
            end
        else CTStatus = 'Change in created time';
        end
    else CTStatus = 'Change in created time';
    end
else CTStatus = 'Change in created time';
end

% check the sensor data value on the file (CREATE USER)
DCU21 = strcmp(Createuser2,Createuser1);
DCU32 = strcmp(Createuser3,Createuser2);
DCU43 = strcmp(Createuser4,Createuser3);
DCU54 = strcmp(Createuser5,Createuser4);
% detect different in Createuser. (1 identical, 0 diff)
if DCU21 == 1 % DCU21=1, no diff between createuser1&2, move on to
check DCU32
    if DCU32 == 1 % DCU32=1, no diff between createuser2&3, move on to
check DCU43
        if DCU43 == 1 % DCU43=1, no diff between createuser3&4, move on
to check DCU54
            if DCU54 == 1 % DCU54=1, no diff between createuser4&5,
report final status
                CUStatus = 'No change in created user';
            else CUStatus = 'Change in created user';
            end
        else CUStatus = 'Change in created user';
        end
    else CUStatus = 'Change in created user';
    end
else CUStatus = 'Change in created user';
end

% check the sensor data value on the file (ACCESS TIME)
```

```
DAT21 = strcmp(Accesstime2,Accesstime1);
DAT32 = strcmp(Accesstime3,Accesstime2);
DAT43 = strcmp(Accesstime4,Accesstime3);
DAT54 = strcmp(Accesstime5,Accesstime4);
% detect different in Accesstime. (1 identical, 0 diff)
if DAT21 == 1 % DAT21=1, no diff between Accesstime1&2, move on to
check DAT32
    if DAT32 == 1 % DAT32=1, no diff between Accesstime2&3, move on to
check DAT43
        if DAT43 == 1 % DAT43=1, no diff between Accesstime3&4, move on
to check DAT54
            if DAT54 == 1 % DAT54=1, no diff between Accesstime4&5,
report final status
                ATStatus = 'No change in accessed time';
            else ATStatus = 'Change in accessed time';
            end
        else ATStatus = 'Change in accessed time';
        end
    else ATStatus = 'Change in accessed time';
    end
else ATStatus = 'Change in accessed time';
end

% check the sensor data value on the file (ACCESS USER)
DAU21 = strcmp(Accessuser2,Accessuser1);
DAU32 = strcmp(Accessuser3,Accessuser2);
DAU43 = strcmp(Accessuser4,Accessuser3);
DAU54 = strcmp(Accessuser5,Accessuser4);
% detect different in Accessuser. (1 identical, 0 diff)
if DAU21 == 1 % DAU21=1, no diff between Accessuser1&2, move on to
check DAU32
    if DAU32 == 1 % DAU32=1, no diff between Accessuser2&3, move on to
check DAU43
        if DAU43 == 1 % DAU43=1, no diff between Accessuser3&4, move on
to check DAU54
            if DAU54 == 1 % DAU54=1, no diff between Accessuser4&5,
report final status
                AUStatus = 'No change in accessed user';
            else AUStatus = 'Change in accessed user';
            end
        else AUStatus = 'Change in accessed user';
        end
    else AUStatus = 'Change in accessed user';
    end
else AUStatus = 'Change in accessed user';
end

% check the sensor data value on the file (MODIFY TIME)
DMT21 = strcmp(Modifytime2,Modifytime1);
DMT32 = strcmp(Modifytime3,Modifytime2);
DMT43 = strcmp(Modifytime4,Modifytime3);
DMT54 = strcmp(Modifytime5,Modifytime4);
% detect different in Modifytime. (1 identical, 0 diff)
if DMT21 == 1 % DMT21=1, no diff between Modifytime1&2, move on to
check DMT32
```

```
    if DMT32 == 1 % DMT32=1, no diff between Modifytime2&3, move on to
check DMT43
        if DMT43 == 1 % DMT43=1, no diff between Modifytime3&4, move on
to check DMT54
            if DMT54 == 1 % DMT54=1, no diff between Modifytime4&5,
report final status
                MTStatus = 'No change in modifyed time';
            else MTStatus = 'Change in modifyed time';
            end
        else MTStatus = 'Change in modifyed time';
        end
    else MTStatus = 'Change in modifyed time';
    end
else MTStatus = 'Change in modifyed time';
end

% check the sensor data value on the file (MODIFY USER)
DMU21 = strcmp(Modifyuser2,Modifyuser1);
DMU32 = strcmp(Modifyuser3,Modifyuser2);
DMU43 = strcmp(Modifyuser4,Modifyuser3);
DMU54 = strcmp(Modifyuser5,Modifyuser4);
% detect different in Modifyuser. (1 identical, 0 diff)
if DMU21 == 1 % DMU21=1, no diff between Modifyuser1&2, move on to
check DMU32
    if DMU32 == 1 % DMU32=1, no diff between Modifyuser2&3, move on to
check DMU43
        if DMU43 == 1 % DMU43=1, no diff between Modifyuser3&4, move on
to check DMU54
            if DMU54 == 1 % DMU54=1, no diff between Modifyuser4&5,
report final status
                MUStatus = 'No change in modifyed user';
            else MUStatus = 'Change in modifyed user';
            end
        else MUStatus = 'Change in modifyed user';
        end
    else MUStatus = 'Change in modifyed user';
    end
else MUStatus = 'Change in modifyed user';
end

% check the sensor data value on the file (FILE SIZE)
DFS21 = strcmp(Filesize2,Filesize1);
DFS32 = strcmp(Filesize3,Filesize2);
DFS43 = strcmp(Filesize4,Filesize3);
DFS54 = strcmp(Filesize5,Filesize4);
% detect different in Filesize. (1 identical, 0 diff)
if DFS21 == 1 % DFS21=1, no diff between Filesize1&2, move on to check
DFS32
    if DFS32 == 1 % DFS32=1, no diff between Filesize2&3, move on to
check DFS43
        if DFS43 == 1 % DFS43=1, no diff between Filesize3&4, move on
to check DFS54
            if DFS54 == 1 % DFS54=1, no diff between Filesize4&5,
report final status
                FSStatus = 'No change in file size';
            else FSStatus = 'Change in file size';
            end
        else FSStatus = 'Change in file size';
        end
    else FSStatus = 'Change in file size';
    end
else FSStatus = 'Change in file size';
end

%SUMMARY of state
Itime = TimeStamp
display(FSStatus)
display(CTStatus)
display(CUStatus)
display(ATStatus)
display(AUStatus)
display(MTStatus)
display(MUStatus)

Identifywrite(Itime,FSStatus,CTStatus,CUStatus,ATStatus,AUStatus,MTStat
us,MUStatus)
```

APPENDIX C. IDENTIFY WRITE XML MATLAB CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identifywrite XML
%
% Edmund Teo
% Version 1.0
% created: 20 Feb 2014
% last mod: Nil
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function
Identifywrite(itime,FSStatus,CTStatus,CUStatus,ATStatus,AUStatus,MTStat
us,MUStatus)

%Write/update the collected identify state into feature matrix
docNode=com.mathworks.xml.XMLUtils.createDocument('Identify');
Identify=docNode.getDocumentElement; % Identify is the root node
Identify.setAttribute('version','1.0');

% Isample header 1
header=docNode.createElement('Isample');
%Isample.setAttribute('1');
Identify.appendChild(header);

% Identify time child
anitem=docNode.createElement('Itime');
anitem.appendChild(docNode.createTextNode(itime));
header.appendChild(anitem);

% Status of file size child
anitem=docNode.createElement('FSStatus');
anitem.appendChild(docNode.createTextNode(FSStatus));
header.appendChild(anitem);

% Status of created time child
anitem=docNode.createElement('CTStatus');
anitem.appendChild(docNode.createTextNode(CTStatus));
header.appendChild(anitem);

% Status of created user child
anitem=docNode.createElement('CUStatus');
anitem.appendChild(docNode.createTextNode(CUStatus));
header.appendChild(anitem);

% Status of accessed time child
anitem=docNode.createElement('ATStatus');
anitem.appendChild(docNode.createTextNode(ATStatus));
header.appendChild(anitem);

% Status of accessed user child
anitem=docNode.createElement('AUStatus');
```

```
anitem.appendChild(docNode.createTextNode(AUStatus));
header.appendChild(anitem);

% Status of modified time child
anitem=docNode.createElement('MTStatus');
anitem.appendChild(docNode.createTextNode(MTStatus));
header.appendChild(anitem);

% Status of modified user child
anitem=docNode.createElement('MUStatus');
anitem.appendChild(docNode.createTextNode(MUStatus));
header.appendChild(anitem);

%Update feature matrix
xmlwrite('D:\Test\Image18022014073000-FM-I.xml',docNode);
type('D:\Test\Image18022014073000-FM-I.xml');
```

LIST OF REFERENCES

Ammon, Grant P. (MC1). “Professional Dialogue with NPS President Helps Students Set Thesis Topics.” July 3, 2013. http://www.nps.edu/About/News/Professional-Dialogue-With-NPS-President-Helps-Students-Set-Thesis-Topics.html.

Ball, Michael. 2013, January 15. “Data Loss Prevention: A Layered Approach Is Best.” http://security-musings.blogspot.com/2013/01/data-loss-prevention-layered-aproach-is.html.

Buede, Dennis. 2009. *The Engineering Design of Systems, Models and Methods,* 2nd ed. Hoboken, New Jersey: Wiley & Sons.

Deptula, Kendra. 2013. “Automation of Cyber Penetration Testing using the Detect, Identify, Predict, React Intelligence Automation Model.” Master’s thesis, Department of Electrical and Computer Engineering, Naval Postgraduate School.

Eddie. 2012, October 3. “How to get a timestamp (date + time) in MATLAB from Eddie (blog).” Programming Tips and Tricks. http://programming-tips-and-tricks.blogspot.com/2012/10/matlab-timestamp.html.

Falkena, Wouter. 2010. “xml2struct.” MATLAB Central. Updated May 15. http://www.mathworks.com/matlabcentral/fileexchange/28518-xml2struct/content/xml2struct.m.

Fraustino, Julia Daisy, Brooke Liu, and Yan Jin. 2012. “Social Media Use during Disasters: A Review of the Knowledge Base and Gaps,” Final Report to Human Factors/Behavioral Sciences Division, Science and Technology Directorate, U.S. Department of Homeland Security. College Park, MD: START.

GitHub. 2014. “Twitter Sample Payload, JSON format.” https://gist.github.com/gnip/764239.

Goshorn, Deborah. SE3201: Engineering Systems Conceptualization, Smart Clouds, using Smart Apps for Tactical Applications. Lecture, Department of Systems Engineering, Naval Postgraduate School, January, 2013.

———. “The Systems Engineering of a Secure Network-Centric Distributed Intelligent System of Systems for Robust Human Behavior Classifications.” Ph.D. Dissertation, Department of Computer Science, UCSD, 2010.

Goshorn, Deborah and Rachel Goshorn. *Cybersecurity in a Network-Centric Smart-Environment System of Systems*. Naval Postgraduate School Cyber Summit, October 29, 2009.

Goshorn, Rachel, Deborah Goshorn, Joshua Goshorn, and Lawrence Goshorn. 2010. “Behavior Modeling for Detection, Identification, Prediction, and Reaction (DIPR) in AI Systems Solutions.” In *Handbook of Ambient Intelligence and Smart Environments*, ed. Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto, 669–700. New York: Springer.

Goshorn, Rachel, Deborah Goshorn, Joshua Goshorna, and Lawrence Goshorn. 2011. “The Need for Distributed Intelligence Automation Implemented through Four Overlapping Approaches: Intelligence Automation Software, Standardization for Interoperability, Network-Centric System of Systems Infrastructure (with Advanced Cloud Computing) and Advanced Sensors.” Center of Excellence in Command, Control, Communications, Computing and Intelligence, George Mason University. http://c4i.gmu.edu/events/reviews/2011/papers/12-Goshorn-paper.pdf.

GSA. 2014. “FebRAMP.” U.S. General Services Administration. Last modified March 25. http://www.gsa.gov/portal/category/102371.

Hamel, Stephane. 2012. “Big Data–What it means for the Digital Analyst.” Online Behavior. http://online-behavior.com/analytics/big-data.

Infocomm Development Authority of Singapore. 2012. “Publication of Infocomm Technology Roadmap 2012.” http://www.ida.gov.sg/~/media/Files/Infocomm%20Landscape/Technology/TechnologyRoadmap/SocialMedia.pdf.

Jewish Business News. 2014, February 13. “Israel Aerospace Industries (IAI) Launches Cyber Early Warning R&D Center In Singapur.” *Jewish Business News*. http://jewishbusinessnews.com/2014/02/13/israel-aerospace-industries-iai-launches-cyber-early-warning-rd-center-in-singapur/.

Jurjonas, B. 2012. “Smart Selection and Configuration of Cyber Sensors for Active Defensive Cyber Operations.” Master’s thesis, Department of Electrical and Computer Engineering, Naval Postgraduate School.

Lappin, Yaakov. 2014, February 13. “IAI Opens Cyber R&D Center in Singapore.” *The Jerusalem Post*. http://www.jpost.com/Defense/IAI-opens-cyber-R-and-D-center-in-Singapore-341294.

Lim, Rachael. 2013, June 30. “New Hub to Defend against Cyber Threats.” *Cyberpioneer*, Ministry of Defence, Singapore. http://www.mindef.gov.sg/imindef/resourcelibrary/cyberpioneer/topics/articles/news/2013/jun/30jun13_news2updated.html#.UmbI4xDAbBd.

Mills, A., R. Chen, J. Lee, and H. R. Rao. 2009. “Web 2.0 Emergency Applications: How Useful can Twitter Be for an Emergency Response?” http://denman-mills.net/web_documents/jips_mills.etal._2009.07.22_finalsubmission.pdf.

Mims, Christopher. 2010, June 14. “How Twitter Helps in a Disaster.” MIT Technology Review. http://www.technologyreview.com/view/419368/how-twitter-helps-in-a-disaster/.

Norton Internet Security unofficial blog. 2011, March 12. “Data Loss Prevention.” http://www.nortoninternetsecurity.cc/2011/03/data-loss-prevention.html).

Onuskanich, Rebecca. 2011. “Out with the DIACAP, In with the DIARMF.” Lunarline Inc. http://www.lunarline.com/sites/default/files/out%20with%20diacap %20in%20with%20diarmf%20-%20lunarline%20white%20paper_dec%2011.pdf.

Perng, S.-Y., R. Halvorsrud, M. Buscher, M. Stiso, L. Wood, L. Ramires, and A. Al-Akkad. 2012. “Peripheral Response: Microblogging during the 22/7/2011 Norway Attacks.” *Proceedings of the 9th International ISCRAM Conference*, Vancouver, Canada. April 2012. L. Rothkrantz, J. Ristvej and Z. Franco, eds. Simon Fraser University, Vancouver, Canada.

Rock Publicity. 2012. “The State of Social Media in Singapore.” http://rockpublicity.com/wp-content/uploads/2012/11/2012-RP-SINGAPORE-SOCIAL-MEDIA-STUDY.pdf.

SANS Institute. 2014. “The Critical Security Controls.” Accessed March 26. http://www.sans.org/critical-security-controls/.

Twitter Developers. 2012. “Things Every Developer Should Know.” Updated April 23. https://dev.twitter.com/docs/things-every-developer-should-know.

University of Miami School of Medicine. 2006. “Privacy/Data Protection Project.” Updated on April 24, 2006. http://privacy.med.miami.edu/glossary /xd_confidentiality_integrity_availability.htm.

United States Southern Command (USSOUTHCOM). 2014. “Contingency Response | Disaster Relief | Humanitarian Assistance.” Accessed March 28. http://southcom.mil/ourmissions/Pages/Contingency-Response--Disaster-Relief--Humanitarian-Assistance-.aspx

W3schools. 2014. “XML Tutorial.” http://www.w3schools.com/xml/default.asp.

Young, Charles P. “Paul” (Lt Col). 2011. “DoD Cyber Operations.” *U.S. Cyber Command. Presented at DoD Enterprise Architecture Conference 2011*, Hampton, VA, May 2. http://www.dodenterprisearchitecture.org /pastmeetings/Documents/2%20Young.pdf.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California